DYNAMIC ADJUSTMENT OF LANGUAGE MODELS FOR AUTOMATIC SPEECH RECOGNITION USING WORD SIMILARITY

Anna Currey, Irina Illina, Dominique Fohr

Université de Lorraine, LORIA, UMR 7503, Vandoeuvre-lès-Nancy, F-54506, France
Inria, Villers-lès-Nancy, F-54600, France
CNRS, LORIA, UMR 7503, Vandoeuvre-lès-Nancy, F-54506, France

ABSTRACT

Out-of-vocabulary (OOV) words can pose a particular problem for automatic speech recognition (ASR) of broadcast news. The language models (LMs) of ASR systems are typically trained on static corpora, whereas new words (particularly new proper nouns) are continually introduced in the media. Additionally, such OOVs are often content-rich proper nouns that are vital to understanding the topic. In this work, we explore methods for dynamically adding OOVs to language models by adapting the n-gram language model used in our ASR system. We propose two strategies: the first relies on finding in-vocabulary (IV) words similar to the OOVs, where word embeddings are used to define similarity. Our second strategy leverages a small contemporary corpus to estimate OOV probabilities. The models we propose yield improvements in perplexity over the baseline; in addition, the corpus-based approach leads to a significant decrease in proper noun error rate over the baseline in recognition experiments.

***Index Terms*—** ASR, language modeling, OOV, word embeddings, lexicon extension

1. INTRODUCTION

Automatic speech recognition (ASR) systems are often trained on large but static text corpora and with a fixed vocabulary. For a system whose goal is to recognize speech about current events, this can pose a problem, since new words are continually introduced based on the events that occur. A particular issue is proper nouns (PNs): the names of newly important people or locations may not be in the vocabulary of the system, but recognizing them can be paramount to understanding the topic. It is not possible to simply find a large enough corpus to cover all of the important words, as novel names will always be introduced into a language. Therefore, a competent ASR system dealing with current events should accommodate adding new words to its vocabulary dynamically. Updating the n-gram language model (LM) of a deep neural network (DNN) ASR system[1] by adding proper nouns to its vocabulary and estimating their parameters is the focus of this article.

Two main strategies for updating n-gram language models with domain-relevant out-of-vocabulary (OOV) words have been proposed in the literature: directly estimating parameters of n-grams containing the relevant OOVs, and class-based adaptation. Direct estimation approaches typically rely on a notion of similarity between OOV and in-vocabulary (IV) words to model OOV behavior. For example, Lecorvé, Gravier, and Sébillot [1] defined equivalence relations for words and for n-grams based on part-of-speech and semantic relatedness. Additionally, Qin [2] suggested generating n-grams to add to a language model by taking an OOV's context and replacing the surrounding IVs with other similar IVs. In class-based adaptation, an OOV word is assigned to a word class and LM probabilities are estimated based on this classification [3]. Extensions to the class-based adaptation model include using linguistic information to better inform classes [4] and using word vectors to cluster OOV words into classes [5]. In addition, Martins, Teixeira, and Neto [6] experimented with updating a language model without any adaptation data. They achieved this by classifying unseen words using morpho-syntactic information.

In this work, we make a number of contributions to the task of dynamically adding OOV words to a language model without retraining the model. Our approaches are based on the idea of directly adding n-grams to the language model and estimating their parameters. First, we expand on the work done in [1] by using word embeddings to find similar IV words after which to model the behavior of the OOV. This allows us to use a similarity measure based on unlabeled text without prior knowledge. Second, we propose using a small corpus contemporary to the test data in order to inform estimates of the behavior of the OOV PNs. This method has the advantage of being easily applicable in a real-world setting, where textual news articles from a given time period could aid in recognition of news from that time period. In addition

[1] State-of-the-art DNN ASR systems use an n-gram LM in combination with a recurrent neural network (RNN) LM.

to being practical, this corpus-based method is also successful, yielding a statistically significant improvement in proper noun error rate over the baseline LM. Notably, the methods proposed in this paper are applied as part of a larger system in which many more OOV PNs are added to the language model than appear in the articles; these methods prove successful in handling this unique challenge.

2. METHODOLOGY

The overall process for recognizing OOV PNs in a news article using our system is as follows: first, the original ASR system is run to create a recognition hypothesis for the article. Based on that initial hypothesis, a list of OOV PNs that are likely to be in the article is generated [7, 8, 9]; the ranking of the OOV PNs is based on topic models using Latent Dirichlet Allocation. These new OOV PNs are added to the lexicon and to the language model, with the parameters of the language model being re-estimated to accommodate the new words. Finally, the system is re-run with the new language model.

The focus of this paper is dynamic re-estimation of the bigram LM used to produce the word lattice of the system. Our input is a baseline LM, a list of OOVs to be added for each article to be recognized, and a word-word similarity table. For some approaches, a corpus of textual data contemporary to the article is also used; we refer to this as the *contemporary corpus*.

From this input, our goal is to apply our estimation approaches in order to create a new LM that contains the vocabulary of the original LM, plus the added OOVs. This includes the following aspects:

- Unigram probability estimation for the OOVs ($P(OOV)$)
- Backoff weight estimation for the OOVs ($B(OOV)$)
- Finding bigrams containing OOVs (*x-OOV* and *OOV-x*)
- New bigram probability estimation ($P(OOV|x)$ and $P(x|OOV)$)

We propose two sets of approaches. *Similarity-based approaches* take advantage of word similarity measures to find similar IVs to a PN OOV; the behavior of the OOV is then modeled after the behavior of the similar IVs. *Corpus-based approaches* use the contemporary corpus to inform language model estimation. The behavior of the OOVs is based on their behavior in the contemporary corpus. Note that corpus-based approaches use word similarity as well, although the basis of these approaches is the contemporary corpus. Similarity-based approaches, on the other hand, do not use any outside corpus at language model estimation time (although outside data is used to create the pre-trained word similarity measure).

In all of our approaches, we use IV words that are similar to a given OOV to inform our LM estimates. In our experiments, our word similarity measures are based on word embeddings trained using the skip-gram architecture with a context window size of two words[2] proposed by Mikolov [10]. We make the assumption that the OOV PNs that we want to add are present in the corpus used for training the embeddings. For instance, a Wikipedia dump could be used in practice as it contains a huge number of (recent) PNs. Given the resulting vector representations of each word, we define word similarity between two words as the cosine similarity between their vector representations.

2.1. Similarity-based approaches

Similarity-based language model estimation approaches rely only on the behavior of similar words; no contemporary corpus is used. This is based on the hypothesis that an added OOV will behave more or less like the existing IVs that are most similar to it; here, we call such words *similar IVs*. Hence, we use the probabilities of n-grams containing those words to estimate the probabilities of n-grams containing the added OOV.

All of the proposed similarity-based approaches use the same overall process to estimate the language model parameters. For each added OOV:

1. Find the similar IVs to the OOV
2. Estimate unigram $P(OOV)$ using unigram probabilities of similar IVs
3. Estimate $B(OOV)$ based on backoffs of similar IVs
4. Choose a similar IV to use for OOV bigram estimation
5. Find all bigrams containing the similar IV; in some cases, select from among those bigrams
6. Add the new bigrams to the language model with the same probabilities, replacing the similar IV with the OOV

Finally, once each OOV is added, the model is renormalized.

2.1.1. Unigram probability estimation

We study the following methods for estimating $P(OOV)$ based on similar IVs, where *closestIV* is the most similar IV word to a given added OOV:

- Closest IV: $P(OOV) \leftarrow P(closestIV)$
- Maximum: $P(OOV)$ is the maximum probability of similar IVs
- Median: $P(OOV)$ is median probability of similar IVs

[2] Small context window size is well-suited to modeling syntactic relations.

In our experiments, we use the five most similar IVs for the maximum and median estimation methods. Note that when the maximum method is used, a single similar IV gives its probability to the OOV; we refer to this IV as the *IV used*.

2.1.2. Estimation of backoff weights

Since the original language model uses modified Kneser-Ney smoothing, the original LM backoff weight for a given word depends on its bigrams. As a result, for estimating backoff weight for an OOV, we simply use the backoff weight of the IV used to find the OOV's bigrams (see below for how this IV is chosen). In addition, only the backoff weights of OOVs are changed; no existing backoff weights are modified.

2.1.3. Finding bigrams

We choose a single IV for each added OOV and model its bigrams after the bigrams of that IV. We study using the closest IV and the IV used to estimate unigram probability (when the unigram probability is defined using the maximum method).

In order to avoid adding too much noise to the adapted language model, we also study limiting the number of bigrams added for each OOV. For each OOV, we choose the M highest-probability bigrams to add to the language model; if M bigrams were not found for that OOV, we add all of the bigrams available.

Note that the bigrams added to the language model are kept separate from the bigrams in the original language model until all bigrams are found. So all new bigrams contain exactly one IV and one OOV. No bigrams containing two OOVs are added to the language model in the similarity-based approaches.

2.1.4. Bigram probability estimation

Each new bigram is based on a unique existing bigram, in which the OOV replaced an IV. Hence, the new bigram is assigned the probability of the corresponding existing bigram.

2.2. Corpus-based approaches

The data-driven word vectors that form the basis of our similarity-based approaches may be unreliable for the added OOV PNs, since these PNs were likely relatively rare in the corpus used to estimate the vectors. Our corpus-based approaches have the advantage of using both the word similarity measures and temporally relevant data (the contemporary corpus) to adapt the language model.

Corpus-based LM estimation is done as follows:

1. Calculate occurrence statistics of the OOV PNs in the contemporary corpus
2. Extract bigrams containing the added OOVs from the contemporary corpus
3. For each such OOV:
   (a) Find the similar IVs
   (b) Use the corpus statistics to estimate $P(OOV)$
   (c) Estimate $B(OOV)$ based on similar IV behavior
   (d) Select contemporary corpus bigrams containing the OOV to add to the LM
   (e) Estimate bigram probabilities using information about similar IVs and corpus statistics
4. Renormalize the language model

2.2.1. Unigram probability estimation

In our experiments, we adapt the baseline language model rather than the original language model[3]; an important consequence is that all of the added OOVs have an initial (very small) probability in the unadapted language model. Therefore, we propose using information about the behavior of the OOV PNs in the contemporary corpus to inflate their baseline probabilities ($P_{base}(OOV)$).

Specifically, we study two ways of defining unigram probabilities for the OOVs, where $N(x)$ is the count of x in the contemporary corpus:

- Maximum likelihood:
  $P(OOV) \leftarrow \max\left(P_{base}(OOV), \frac{N(OOV)}{\sum_w N(w)}\right)$
- Weighted by number of corpus occurrences of the OOV

2.2.2. Estimation of backoff weights

Backoff weights of the added OOVs ($B(OOV)$) are assigned based on the backoff weights of the closest IVs or of <unk>. We do not attempt to train backoff weights using the corpus, as we expect overall behavior of the OOVs to be similar to that of the similar IVs or of <unk>, whereas training reliable Kneser-Ney backoff weights from the corpus would likely require much more information.

The methods for assigning backoff weights we study are:

- <unk>backoff: $B(OOV) \leftarrow B(\text{<unk>})$
- Closest IV backoff: $B(OOV) \leftarrow B(closestIV)$

2.2.3. Finding bigrams

One of the major advantages of having access to the contemporary corpus is that the corpus can be used to find bigrams. Our initial proposal for doing this is as follows: for each OOV, add a bigram containing that OOV if and only if that bigram was found in the corpus. These bigrams can be interpreted as

[3] The baseline LM is created from the original LM by giving a small probability to each added OOV. We adapt the baseline to ensure that all adapted LMs have the same vocabulary, allowing for direct perplexity comparisons.

safe choices: since they actually occur in the contemporary corpus, they may occur in the test data.

We study the following methods for using the corpus to find bigrams for a given added OOV:

- All corpus bigrams: add all bigrams from the corpus containing the OOV
- Limiting bigrams: only add bigrams that occur in the corpus more than a given amount of times

The limiting bigrams approach allows us to add only bigrams about which we are quite confident, while the all corpus bigrams approach allows us to increase bigram coverage.

Note that when a bigram containing an added OOV PN and an OOV that is not added to the LM is found in the contemporary corpus, it is ignored, since we cannot add such bigrams to the language model. However, we do add bigrams consisting of two added OOV PNs to the LM; we call these *OOV-OOV* bigrams. Indeed, we do not expect this to be an edge case, since many of the added OOVs are names of people; the first and last names of a person often appear together.

2.2.4. Bigram probability estimation

We divide bigram probabilities into those of the form $P(x|OOV)$ (including *OOV-OOV* bigrams) and those of the form $P(OOV|x)$. We consider two methods for estimating $P(x|OOV)$. The first consists of giving uniform probabilities to each bigram with the same first word. In the second method, we weight the bigram probability by the number of occurrences of the bigram in the contemporary corpus.

$P(OOV|x)$ bigram probabilities can be estimated using information about the behavior of existing bigrams with the same first word, *x*. We refer to such existing bigrams as bigrams of the type *x*-*y* (where *y* is an IV), and to their probabilities as $P(y|x)$. The following approaches to estimating $P(OOV|x)$ are studied:

- Minimum: $P(OOV|x) \leftarrow \min_{y:P(y|x)\neq 0} P(y|x)$
- Closest IV: $P(OOV|x) \leftarrow$ $P(\text{argmax}_{y:P(y|x)\neq 0}\, similarity(y, OOV)|x)$
- Maximum of most similar IVs: $P(OOV|x) \leftarrow$ $max_{simIVs:P(simIV|x)\neq 0}P(simIV|x)$, where *simIV* means similar IV

In the last case, we take the five most similar IVs such that $P(simIV|x) \neq 0$.

3. EXPERIMENTAL SETUP

3.1. Data

3.1.1. Training corpora

Training data was required to create the original language model, the word embeddings, and the lists of OOV PNs to add. For each of these tasks, some combination of the following three training sets was used:

- *Le Monde* + *Gigaword*: textual data from the French newspaper *Le Monde* and from the French Gigaword corpus (1B words; published between 1994 and 2008)
- *Le Figaro*: textual data from the French newspaper *Le Figaro* (8M words; 2014)
- *L'Express*: textual data from the French newspaper *L'Express* (51M words; 2014)

The original language model was trained using the *Le Monde* + *Gigaword* corpus. The word embeddings were created using a concatenation of the three corpora. The lists of OOV PNs to add were created using the *L'Express* corpus. The *Le Figaro* corpus was used as the contemporary corpus for corpus-based estimation, corresponding to the same time period as the development and test data.

3.1.2. Development corpus

The development corpus comes from the website of the news channel Euronews[4]. It consists of 1,962 textual news articles (510,351 words) from January 2014 to June 2014. This corpus is used to evaluate each of our proposed approaches and select the approaches for which to run the test perplexity and recognition experiments. The OOV rate is 2.7% (13,768 words).

3.1.3. Test corpus

Once we have found the best parameter configurations and algorithms on the development corpus, we use the test corpus to further examine their performance. The test corpus consists of video reports from the Euronews website and their accompanying transcripts. The video reports are used for the recognition experiments, while the transcripts are used for the perplexity experiments. It is important to note that the reference transcriptions for the recognition experiments are the transcripts provided with the news videos, which may not always be an exact match to the audio. The articles were published in the first half of 2014. The test corpus consists of 467 articles (91,880 words), and the OOV rate is 2.3% (2,074 words).

3.2. Kaldi-based Automatic Transcription System

The Kaldi-based Automatic Transcription System (KATS) uses context dependent DNN-HMM phone models trained on 200-hour broadcast news audio files. The lexicon contains about 96K words. Using the SRILM toolkit [11], a bigram language model is estimated on the *Le Monde* + *Gigaword* corpus and used to produce the word lattice.

[4] http://fr.euronews.com/

3.3. Language models

We use the *original* language model of the system, a *baseline* model, and an *oracle* model in our experiments. The *original* LM is a bigram model with modified Kneser-Ney smoothing trained on the *Le Monde* + *Gigaword* data.

Since we add words to the vocabularies of the adapted models, these models do not have the same vocabulary as the original LM. This means that we cannot compare the perplexities of these models with the perplexity of the original LM. As a result, we create the *baseline* language model, which has the same vocabulary as the adapted models. The baseline LM is created from the original LM by adding the words from the list of OOV PNs to be added. Only unigrams are added to the original LM to create the baseline. The probability mass assigned to the new unigrams is taken directly from <unk >, so renormalization is not required.

The *oracle* model is also created as a point of comparison to the adapted models. It represents the upper limit of how much the adapted models can be improved using the contemporary corpus. Like for the baseline model, the vocabulary of the oracle model consists of the vocabulary of the original language model plus the words in the list of OOVs to add. Our oracle model is an interpolation of the original LM and a language model estimated on the contemporary corpus.

3.4. Word lists

For each article, a ranked list of OOV PNs to be added to the LM was generated [9]. We used the 128 words with the highest rankings to create the per-article added OOV lists. An automatic grapheme-to-phoneme conversion was used to assign pronunciations to the OOVs [12].

The per-article added OOV lists are combined to form a list for the entire development corpus and one for the entire test corpus. The development OOV list contains 8,066 words, while the test OOV list contains 6,382 words.

4. RESULTS

We studied each of the proposed methods on the development corpus; the best-performing methods were then applied on the test corpus. Note that due to limitations in our original ASR system, we did not create per-article LMs[5]. Instead, we created a single LM for the whole development corpus and one for the whole test corpus for each method.

4.1. Results on the development corpus

For efficient experimentation, we divided adaptation into subtasks. For each subtask, we found the method that yielded the best result in terms of perplexity; we then used that method on all subsequent experiments.

Table 1 shows the perplexities (PPL) on the development corpus when each of the unigram probability estimation methods was applied. Weighting the baseline probability by the number of corpus occurrences performed best overall, while taking the maximum probability of the similar IVs performed best among the similarity-based methods. However, we should note that there did not seem to be a large difference between the similarity-based methods. In subsequent experiments with similarity-based approaches, we estimated unigrams using maximum probability; for corpus-based approaches, we used weighting by corpus occurrences.

	Unigram estimation method	**PPL**
	baseline	230.4
	oracle	214.0
similarity-based	closest IV probability	229.1
	maximum probability	**228.2**
	median probability	229.0
corpus-based	maximum likelihood	228.0
	weighted by corpus occur.	**227.2**

Table 1. Perplexity results for unigram probability estimates on the development corpus.

Tables 2 and 3 display the perplexity results on the development corpus using similarity-based and corpus-based bigram estimation, respectively. Note that, in the similarity-based approaches (table 2), we chose six and 24 bigrams to add per OOV because there were an average of six unique bigrams for each OOV in the contemporary corpus, while the oracle model added an average of 24 bigrams per added OOV.

	Bigram estimation method	**PPL**
	baseline	230.4
	oracle	214.0
	unigrams only	228.2
finding bigrams	bigrams of closest IV	**231.0**
	bigrams of IV used	232.3
bigrams per OOV	all available bigrams added	231.0
	6 bigrams added per OOV	228.3
	24 bigrams added per OOV	**228.2**

Table 2. Perplexity results for similarity-based bigram estimation on the development corpus.

From table 3, we can see that our best proposed corpus-based method yielded a large improvement over the baseline; the reduction in perplexity was 40.4% of the highest possible reduction (defined using the perplexity of the oracle model). In contrast to the similarity-based bigram methods in table 2, the corpus-based bigram methods resulted in an improvement over adding just unigrams to the LM (see table 1 for comparison).

[5] Computing the Finite-State Transducer (FST) for each article to be recognized is too time consuming.

	Bigram estimation method	**PPL**
	baseline	230.4
	oracle	214.0
	unigrams only	227.2
finding bigrams	all corpus bigrams	227.4
	limiting bigrams – cutoff 2	227.4
	limiting bigrams – cutoff 5	**227.3**
	limiting bigrams – cutoff 10	227.3
backoff	<unk >backoff	227.3
	closest IV backoff	**226.8**
$P(x\|OOV)$	uniform	**226.8**
	weighted by corpus occur.	230.1
$P(OOV\|x)$	minimum	226.8
	using closest IV	225.3
	max. of most similar IVs	**223.8**

Table 3. Perplexity results for corpus-based bigram estimation on the development corpus.

4.2. Results on the test corpus

We selected the best-performing corpus-based and similarity-based methods from the experiments on the development data to be applied to the test data. For the similarity-based method, we used maximum probability for unigrams and added 24 bigrams per OOV. For the corpus-based method, we weighted unigram probabilities by corpus occurrences and assigned closest IV backoff. Bigrams were added with a cutoff of 5 occurrences in the contemporary corpus, with $P(x|OOV)$ uniform and $P(OOV|x)$ assigned as the maximum probability of the most similar IVs.

Since the similarity-based bigram approach did not outperform the similarity-based unigram-only approach on the development data, we included a similarity-based approach in which only unigrams were added to the LM.

Table 4 displays the results of the perplexity experiments on the test data. These results are very similar to what was observed for the development data. All methods improved over the baseline; however, adding unigrams and bigrams using a similarity-based approach did not do as well as just adding unigrams. In addition, the corpus-based method performed better than either similarity-based method.

Method	**PPL**
baseline	212.0
oracle	197.4
similarity-based unigrams	210.2
similarity-based unigrams and bigrams	210.2
corpus-based	**206.6**

Table 4. Perplexity results on the test data.

The results for the recognition experiments on the test corpus are shown in table 5. We display the word error rate (WER) and proper noun error rate (PNER) on the test data for each of the adapted models, as well as for the original, baseline, and oracle language models. Statistical significance was calculated using the matched-pairs significance test described in [13]. These results closely mirror the development experiments and the perplexity results on the test data; namely, the corpus-based approaches outperform the similarity-based approaches and the baseline.

Method	**WER**	**PNER**
original LM	33.8	56.8
baseline	33.4	52.3
oracle	33.1	49.9
similarity-based unigrams	33.4	52.1
similarity-based unigrams and bigrams	33.4	52.2
corpus-based	**33.2***	**50.9***

Table 5. Recognition results on the test data. Asterisks indicate statistically significant improvements over the baseline.

5. CONCLUSIONS

In this work, we set out to explore ways of improving the recognition of OOVs in an ASR system by adding OOVs to the language model. This was done by dynamically updating the LM without retraining it. We proposed two strategies: a similarity-based approach that took advantage of word embeddings to model behavior of added OOVs after that of IVs close to them in the vector space, and a corpus-based approach that used a small corpus from the same time period as the test data to find bigrams and estimate probabilities. The models we proposed yielded improvements in perplexity over the baseline; in addition, the corpus-based approach led to a significant decrease in PNER. This PNER decrease shows that new proper nouns were appropriately added to the language model. Unlike previous work on this topic, we found that using a small contemporary corpus was more successful in defining the LM behavior of the new OOVs than using the behavior of similar IVs.

Acknowledgements

This work was funded by the ContNomina project supported by the French National Research Agency (ANR) under contract ANR-12-BS02-0009, and by an Amazon Academic Research Award.

6. REFERENCES

[1] Gwénolé Lecorvé, Guillaume Gravier, and Pascale Sébillot, “Automatically finding semantically consistent

n-grams to add new words in LVCSR systems," in *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2011, pp. 4676–4679.

[2] Long Qin, *Learning out-of-vocabulary words in automatic speech recognition*, Ph.D. thesis, Carnegie Mellon University, 2013.

[3] Alexandre Allauzen and Jean-Luc Gauvain, "Open vocabulary ASR for audiovisual document indexation," in *Proceedings of the 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2005, vol. 1, pp. 1013–1016.

[4] Aleš Pražák, Pavel Ircing, and Luděk Müller, "Language model adaptation using different class-based models," in *Proceedings of the 2007 International Conference on Speech and Computer*. Springer-Verlag, 2007, pp. 449–454.

[5] Welly Naptali, Masatoshi Tsuchiya, and Seiichi 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.

[6] Ciro Martins, António J.S. Teixeira, and João Paulo Neto, "Automatic estimation of language model parameters for unseen words using morpho-syntactic contextual information," in *Proceedings of the Ninth Annual Conference of the International Speech Communication Association*. ISCA, 2008, pp. 1602–1605.

[7] Imran Sheikh, Irina Illina, Dominique Fohr, and Georges Linarès, "Learning to retrieve out-of-vocabulary words in speech recognition," *arXiv preprint arXiv:1511.05389*, 2015.

[8] Imran Sheikh, Irina Illina, Dominique Fohr, and Georges Linarès, "OOV proper name retrieval using topic and lexical context models," in *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2015, pp. 5291–5295.

[9] Imran Sheikh, Irina Illina, Dominique Fohr, and Georges Linarès, "Document level semantic context for retrieving OOV proper names," in *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2016, pp. 6050–6054.

[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[11] Andreas Stolcke, "SRILM—an extensible language modeling toolkit," in *Proceedings of the Seventh International Conference on Spoken Language Processing*. ISCA, 2002, pp. 901–904.

[12] Irina Illina, Dominique Fohr, and Denis Jouvet, "Grapheme-to-phoneme conversion using conditional random fields," in *Proceedings of the Twelfth Annual Conference of the International Speech Communication Association*. ISCA, 2011, pp. 2313–2316.

[13] Laurence Gillick and Stephen J. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proceedings of the 1989 International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1989, pp. 532–535.