Personalization of Word-Phrase-Entity Language Models

M. Levit, A. Stolcke, R. Subba, S. Parthasarathy, S. Chang, S. Xie, T. Anastasakos, B. Dumoulin

Microsoft Corporation, U.S.A.

mlevit@microsoft.com

10.21437/Interspeech.2015-173

Abstract

We continue our investigations of Word-Phrase-Entity (WPE) Language Models that unify words, phrases and classes, such as named entities, into a single probabilistic framework for the purpose of language modeling. In the present study we show how WPE LMs can be adapted to work in a *personalized* scenario where class definitions change from user to user or even from utterance to utterance. Compared to traditional class-based LMs in various conditions, WPE LMs exhibited comparable or better modeling potential without requiring pre-tagged training material. We also significantly scaled the experimental setup by widening the target domain, amplifying the amount of training material and increasing the number of classes.

Index Terms: Language model personalization, class-based LMs, Word-Phrase-Entity LMs

1. Introduction

Situation-aware modeling of human behavior is necessary to produce high quality human-machine interactions for a variety of tasks and scenarios. In speech recognition, and specifically language modeling, a wide spectrum of techniques have been proposed to dynamically adjust statistical language models to the interaction context: from customized turn specific grammars in traditional telephony IVR, to cache- and trigger-based language models that learn from the immediate past [1, 2], to topic-specific language models that derive topic information from the utterance itself [3, 4]. We are interested in scenarios where the underlying language structure can be considered fixed within the domain, but contents of individual syntactico-semantic slots, such as named entities, vary from case to case, from user to user. Class-based language models is the traditional solution for such scenarios. For personalization purposes, class definitions can be compiled from meta-information associated with the user. Class-based language models view sentences as sequences of class tokens (rather than words) where each token can be realized in a number of different ways. Since there are generally several possible parses $\boldsymbol{c}^k$ to express sentence $\boldsymbol{w}$ in terms of tokens c_i^k with respective instantiations π_i^k, its total probability can be decomposed as follows:

$$P(\boldsymbol{w}) = \sum_k \prod_i P(c_i^k|h_i^k)P(\pi_i^k|c_i^k). \quad (1)$$

where h_i^k is the history of c_i^k in $\boldsymbol{c}^k$. Typically, classes would be either pre-defined or derived from the data by analyzing context similarity of candidate words [5]. In parallel, considerable effort has been invested into finding sequence of words that frequently occur together and turning them into special "phrase" tokens (e.g. [6, 7]). In most realizations, decisions to replace a word or a sequence of words by a class- or phrase token were made greedily for the entire corpus: for instance, once it was determined that several words formed a phrase, all their occurrences would be replaced by the corresponding phrase token.

Recently, we proposed an extension of the class-based paradigm that not only unified classes and phrases in a single probabilistic framework, but also departed from the greedy approach by making individual token-replacement decisions for each candidate group of words [8]. The Maximum-Likelihood approach would start with a word-level corpus and several generic class-definitions (encoded as word tries or finite automata). It would then iterate by alternating ML-parsing of the corpus with the current WPE language models with updating token n-gram statistics constituting these models from the produced parses. A simple extension of this approach allowed for a simultaneous optimization of the class definitions as well. Applied to a limited target scenario, WPE language models significantly reduced perplexity and WER on unseen data.

In the present report we demonstrate that WPE LMs offer themselves as a natural choice for personalized speech recognition. We show how WPE LMs, in comparison with traditional class-based personalized language models trained on pre-tagged training material, achieve similar or slightly better perplexity while making no assumptions on availability of entity-specific taggers. Several speech recognition experiments in various setups are presented to further support our findings regarding usability of personalized WPE LMs.

The rest of this paper is organized as follows. Section 2 introduces WPE LMs. In Section 3 we suggest modifications to the training algorithm to incorporate personalized class definitions and discuss ways to overcome engineering challenges that a developer of such a system would face. Our experimental setup is described in Section 4 followed by discussion of the results. Suggestions for future work and a summary of the most important results conclude the paper.

2. Word-Phrase-Entity Language Models

WPE Language models [8] are used just like regular class-based language models (except for treating a number of word-phrases as special pseudo-words), but their training is carried out in an iterative manner. Each EM iteration consists of two steps:

Expectation: use the current model to produce a collection of alternative parses for each sentence with joint probabilities

$$P(\boldsymbol{w}, \boldsymbol{c}^k) = \prod_{c_i^k \in \boldsymbol{c}^k} P(c_i^k|h_i^k)P(\pi_i^k|c_i^k). \quad (2)$$

This is currently carried out with the trellis decoder from the SRILM toolkit [9].

Maximization: re-estimate token n-gram probabilities from the produced parses as

$$P^{\text{ML}}(c|h) := \sum_{\boldsymbol{w}} \sum_k L'(\boldsymbol{w}, \boldsymbol{c}^k) \left. \frac{\#ch}{\#h} \right|_{\boldsymbol{c}^k} \quad (3)$$

with count contributions $\frac{\#ch}{\#h}\Big|_{\boldsymbol{c}^k}$ from a particular parse $\boldsymbol{c}^k$ of a particular training sentence $\boldsymbol{w}$ being weighted by a combination of relative frequency of the sentence in the corpus and the parse posterior for the sentence:

$$L'(\boldsymbol{w}, \boldsymbol{c}^k) = L'(\boldsymbol{w}) * \frac{P(\boldsymbol{w}, \boldsymbol{c}^k)}{\sum_{\boldsymbol{c}} P(\boldsymbol{w}, \boldsymbol{c})} \tag{4}$$

This step is implemented via standard n-gram LM training and therefore can benefit from discounting methods (that understand fractional weights). Similarly, the algorithm can tune up shared class definitions by updating their internal probabilities with

$$P^{\mathrm{ML}}(\pi|c) = \sum_{\boldsymbol{w}} \sum_{k} L'(\boldsymbol{w}, \boldsymbol{c}^k) \; \frac{\#(c, \pi)}{\#c}\Big|_{\boldsymbol{c}^k} \tag{5}$$

Optimization starts with a simple unigram language model based on counts of all word n-grams (up to a given length) in the training corpus. During our experiments we have found out that certain heuristic adjustments such as scaling joint log-probabilities $P(\boldsymbol{w}, \boldsymbol{c})$ by the word length of w improve convergence and can lead to better accuracy on unseen data.

3. Adding Personalization

Language models can be personalized in many ways across different applications such as speech recognition, search and recommendation. For instance, in search, adaptive techniques to predict the applicability of user personal data given the current task and activity context have shown to be effective [10]. In [11], language models trained from the most frequent tags of the users in bookmarks are used to deliver personalized recommendations. For speech recognition, in the space of contextual word probabilities, pre-trained language models can be adapted on the training material collected for a particular user [13] or, for exponential LMs, user affiliation information can be encoded as an additional feature [14]. The range of sources that the personal data can come from is diverse and includes user's historical behavior [12, 14], social environment [13], or even eye gaze [15]. For the experiments in this paper, we restrict the notion of personalization to only substituting class-definitions, such as contact names found in the address book of a mobile user. We call such classes "personal grammars". This extension will have two implications on the optimization process. First, personal grammars will need to be taken into account when computing joint probabilities

$$P(\boldsymbol{w}, \boldsymbol{c}, u) = \prod_{c_i \in \boldsymbol{c}} P(c_i|h_i) P(\pi_i|c_i, u), \tag{6}$$

which is essentially saying that probabilities of class instances (e.g. *"john smith"* in class CONTACT_NAME) are now specific for a particular user u. Formulae (3-5) are adjusted accordingly to employ user-dependent weight $L'(\boldsymbol{w}, \boldsymbol{c}, u)$ and re-estimate updated value for $P^{\mathrm{ML}}(\pi|c, u)$. Second, optimization of the personal grammars will need to happen on a per-user basis. Because of data scarcity, we did not attempt this in our experiments.

The brute force application of the WPE training algorithm would require too much memory or time (an average user has several dozens or even hundreds of items in the grammar). On the other hand, for the purpose of training (and perplexity evaluation), one does not need to load the entire grammars but rather small subsets of them that only contain the words already present in the respective sentences. This trick along with some intelligent dictionary optimization allows to scale training of personalized WPE LMs to millions of sentences and still conduct it on a single CPU.

4. Experimental Setup

One major weakness in the setup of our previous experiments in [8] was the narrow domain (calendar tasks) and a small size of the training corpus (only 20K unique sentences). The present investigation goes beyond these limitations. We remain within the Personal Assistant scenario but remove any domain restrictions. More specifically, our new training corpus consists of about 940K professionally transcribed queries and utterances from users' interactions with Microsoft's automated personal assistant Cortana. The examples cover a wide range of domains from voice search to message dictation, from command and control to chit chat. A pair of validation and test sets from the same source comprising 20K and 5K sentences respectively was also prepared. A separate set of 5K utterances was used only for recognition experiments[1].

It is important to understand that recognition accuracy of a personalized model on natural language sentences actually containing user-specific references (such as contact names) is likely to be underestimated. Indeed reference transcriptions against which the models are evaluated, are generated by human labelers without access to users' personal information. As a result, they tend to suggest common spelling variants for acoustically ambiguous cases (e.g. *"chris"* and not *"kris"*), while the personalized model that actually does have access to user's personal information might expect/suggest correct though less common forms. Therefore, we have devised two alternative testing scenarios to evaluate our personalized WPE LMs. In the first scenario, contrived grammars are generated based on tagged representations of the reference transcriptions and true entity-related statistics from the corpus. In the second scenario, true user grammars are employed.

4.1. Contrived Grammars

We started by analyzing our training and test corpora with a pre-trained sequence tagger designed to recognize named entities (contact names among them) in user requests. The tagger was trained with Conditional Random Fields (CRF) using lexical features such as the identity of the current word, preceding and following words, and associated n-gram features. A small set of regular expressions, and gazetteers for named entities, built by mining various relevant web resources, was used to trigger presence of entity features [16].

For each training/test sentence u, we generate its grammar in four steps: first, a number $s(u)$ of distinct grammar entries is sampled from an empirical distribution histogram $\mathcal{S}$ estimated from real examples (typically less than few hundred entries are expected). If a contact name was hypothesized in the sentence by the tagger, with probability θ we seed the grammar with this hypothesis. For instance, if the tagger marked *"john"* and *"mary"* as contact names in sentence *"tell john that mary is out"*, either of them will be present in the corresponding contrived grammar with probability θ. We also call θ *entity precision*. The remaining entries are sampled from a cumulative distribution of all contact names hypothesized in the entire training set (for instance, *"mom", "my wife"* are likely to have high probabilities in this distribution). Finally, we set weights of

[1]The separation had to be enforced due to a combination of user privacy restrictions and internal implementation details.

all individual entries in the grammar at random, except for the seeds (such as *"john"* and *"mary"* in the example above) whose relative weight is μ times the average. The heuristics $\theta \approx 0.6$ and $\mu \approx 10$ were obtained based on careful evaluation of a few hundred samples from the training corpus with the real personal meta-data (contact names with corresponding use frequencies).

4.2. Realistic Grammars

The realistic personal grammars for contact names are generated per-user based on the user's address book data that is collected with the user's consent. We utilize call and text history counts collected over the recent few months to weigh the items in the personal contacts grammar. For each contact in the address book, we generate items with first name, last name, first and last name, as well as nickname, each associated with their respective call and text history counts. We merge identical items in the list, aggregating the counts in the process and then producing an <item,weight> tuple list. A ceiling function with an upper bound is applied, giving more weight to first name items. The weights for the list of items are then normalized. The weighted list of items is processed to generate the final personal CONTACT_NAME class grammar for a given user. Standard precautions are taken to guarantee confidentiality of all personally identifiable information.

In addition to the personalized CONTACT_NAME grammar, 21 generic named entity classes are used in this setup. This is a significant increase compared to our pilot experiments from [8] where only six classes were used. Their exhaustive list along with sizes and examples (obvious and less obvious) is shown in Table 1. All named entities are initialized as lists of alternatives, except DATE and TIME which are FSMs. Decisions as to which named entities to create classes for were made based solely on data availability, and all class definitions were harvested from publically available online sources, such as Wikipedia, with a small amount of manual adjustments.

Table 1: *Named Entity (NE) classes used in the experiments with realistic grammars.*

NE	size	examples
ACTOR	16585	meryl streep, da brat
MOVIE	10957	forrest gump, talk to her
LASTNAME	9716	jones, summer
CITY	3934	san francisco, franklin
MUSICIAN	3128	beattles, suzanne vega
FIRSTNAME	3036	john, will
MODEL	2204	adriana lima, queen rania
WEBSITE	1610	wikipedia dot org, qq
SONG	1231	stairway to heaven, happy
BUSINESSMAN	1125	larry ellison, estée lauder
ATHLETE	1059	lebron james, kaka
COUNTRY	249	france, us
WORLD_CITY	100	rome, nice
NFL_TEAM	94	new york giants, the bears
MLB_TEAM	87	boston red sox, tigers
NBA_TEAM	82	lakers, bulls
NHL_TEAM	79	boston bruins, kings
STATE	55	california, c a
DAY_OF_WEEK	7	monday
DATE	16st/132arcs	march first two thousand
TIME	11st/72arcs	seven twenty p m

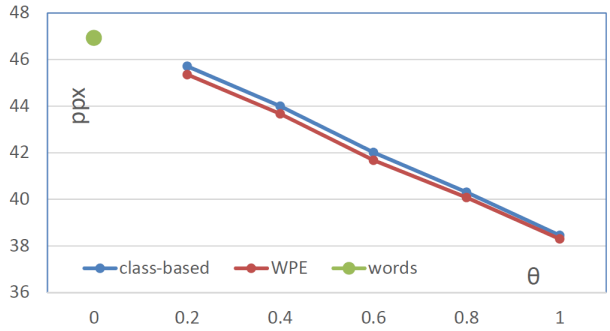

Figure 1: *Test set perplexity as a function of entity precision in the personal grammars; comparison of traditional class-based and WPE LMs.*

5. Experiments and Results

In our first series of experiments with contrived grammars, we compare WPE LMs against traditional class-based language models. No classes except personalized CONTACT_NAME grammars are used in this experiment. Our baseline is a class-based 4-gram LM trained on the automatically tagged training corpus with personal grammars generated as described in Section 4.1. The WPE alternative is a 4-gram token-level LM that allows instances of the personalized class as well as phrases of up to six words with cumulative frequency in the parsed corpus of at least 30. It was trained with 10 iterations of the EM-algorithm as described in [8]. Depending on setup between 3700 and 3900 such phrases ended up in the final WPE LM. While keeping other heuristics fixed, we focus on entity precision θ which can be interpreted as the probability of referencing a contact name that is present in the corresponding grammar definition. Since individual runs exhibit vastly different OOV rates, our perplexity metric assigns the OOV-words unigram probability of 1e-7 which is a conservative estimate only slightly below the lowest unigram probabilities in the training corpus. The results are summarized in Figure 1.

It can be seen that both LM types exhibit similar behavior with a slight but consistent advantage for the WPE language models, perhaps more so in the lower precision range. This can be explained, in part, by the soft class labeling in WPE which tends to preserve entity wording alongside the hypothesized class-name. Another, though closely related, explanation for improved perplexity is how WPE training distributes weight between alternative parses [8]. For instance, while *"mom"* could be in the address book of the user, the command *"call mom"* is so common that it is cheaper to model it via a word bigram or a special phrase "call+mom" and not via class CONTACT_NAME.

However, the most important distinction between the two approaches lies in their expectations regarding training material. While WPE LMs tag the data on their own as we train the language model, to train a traditional class-based LM, a pre-tagged training corpus must be available. It can either be produced manually, which makes scaling prohibitively expensive, especially in the case of personalized class definitions, or it can rely on pre-trained taggers, as in our experiments. Nonetheless, even with the high quality domain-specific tagger, the WPE LM ended up exhibiting better modeling potential.

The rest of the experiments in this paper pertain to the realistic scenario in which actual grammars assembled for each utterance from user's address book are used for personaliza-

tion. First, we would like to see the effect of personalization and generic classes on LM perplexity. This experiment has two baselines: a large general purpose 5-gram language model (GLM) trained on trillions of examples and the aforementioned word level 4-gram LM (LM_0). The traditional class-based baseline was abandoned for this setup, because the latter does not leave room for approximation and would require the entire training set to be manually tagged.

The top part of Table 2 compares perplexities and WERs of three WPE LM versions against the baselines above on the entire test set and on its personalizable and non-personalizable subsets[2]. The first WPE LM (LM_{21}) was built with 21 generic classes whose initial weights in the respective tries and FSMs are optimized in the course of the training process from [8]. The second (LM_1) only has a single class of personal grammars CONTACT_NAME. With 21+1 classes, the last WPE LM (LM_{21+1}) is a combination of the first two. As before, in order to bring all runs down to a common denominator, OOV probabilities are set for perplexity evaluations: 1e-7 for all LMs trained on the in-domain training corpus and 1e-9 for the large general purpose baseline LM.

Table 2: *Test set perplexity, WER(%) and WERR(%) for baseline LMs and WPE configurations in the realistic scenario. For perplexity evaluation, personalizable (PER) and non-personalizable (NPER) subsets are considered separately.*

LM	PER	NPER	total	WER	WERR
GLM	151.16	40.17	46.57	13.83	-
LM_0	136.25	41.06	46.93	16.59	0
LM_{21}	112.68	39.28	44.18	15.58	6.1
LM_1	48.66	41.24	42.02	14.91	10.1
LM_{21+1}	47.13	39.48	40.27	14.62	11.9
LM_0+GLM	109.69	32.25	36.97	13.23	0
LM_{21}+GLM	97.99	31.94	36.19	12.99	1.8
LM_1+GLM	40.82	32.61	33.44	12.11	8.5
LM_{21+1}+GLM	40.38	32.1	32.93	11.97	9.5

Looking at the perplexity numbers, we observe that 21 generic classes (LM_{21}) are helpful for sentences with or without suspected personal references. The effect on the latter can be attributed to the NE classes FIRSTNAME and LASTNAME. On the other hand, the setup with the personal CONTACT_NAME grammars as the only class (LM_1), while slashing perplexity of the personalizable set almost three times, has a slightly adverse effect on the rest of the test corpus. In line with our expectations, the overall perplexity of the LM_1 model matches almost exactly the perplexity of the WPE model in the contrived-grammar setup (see Figure 1) at $\theta = 0.6$, the value gauged from the realistic grammars. The ensemble of all classes (LM_{21+1}) brings about the largest overall perplexity reduction of more than 14%. Similar improvements are achieved for WER on the audio corpus with a total of 11.9% relative reduction to LM_0 (mind, however, that the absolute WERs of the in-domain LMs are higher than might be suggested by perplexity numbers, due to a later time stamp of the audio test set and possible overtraining).

Next, we would like to see how the personalization advantage of the LM_{21+1} propagates through interpolation with the GLM. The interpolation algorithm from [17] for class-based, and in particular WPE LMs, is employed for this purpose. Just like the WPE training in Section 3, this algorithm also had to be adjusted to accommodate personal class definitions. Our in-domain validation set was used to optimize 8.5K context-dependent interpolation weights [18] that preserved cumulative frequency in the corpus of at least 3.0 throughout all EM iterations. The weights were seeded with context-independent values previously tuned-up on the same validation set. The bottom part of Table 2 summarizes perplexities for interpolated versions of all four in-domain LMs (P(OOV)=1e-9). It shows that the WPE advantages for personalized and non-personalized classes are largely preserved even after interpolation with a strong baseline such as our GLM. For instance, measured on all test material, LM_{21+1} had a 11% lower perplexity than LM_0 and 9.5% lower WER.

[2]An example is deemed personalizable if the grammar contains at least one entry whose text can be matched in the sentence.

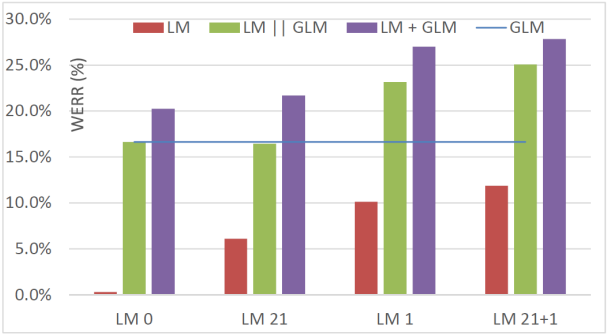

Figure 2: *WER reduction relative to LM_0 for in-domain LMs, and products of their parallel and interpolated combinations with GLM.*

To further validate the conclusion, Figure 2 summarizes WERR of various in-domain LMs and their combinations with the GLM, this time also including empirically best parallel combinations (Kleene closures) LM || GLM. The advantage of the WPE LMs, especially in conjunction with LM interpolation is evident.

Finally, the practical value of the WPE approach needs to be assessed. Indeed, recognition is expected to take longer when a large LM references many moderately sized classes, especially if contents of these classes is dynamic as is the case with personalized grammars. Our measurements revealed that the 21 generic classes added an average of 15% extra recognition time and the personal grammar CONTACT_NAME added 9%. No combination of the two exceeded 30% added latency, which is comparable to a setup where large GLM and medium sized WPE are run in parallel.

6. Future Work and Conclusion

We have demonstrated how Word-Phrase-Entity language models can be successfully used in scenarios that rely on personalized class definitions such as user contact names. With simulated personal class definitions, personalized WPE LMs slightly outperformed traditional class-based LMs while not requiring specially tagged training material. In a setup with realistic class-definitions, WPE LMs using combinations of generic classes with tunable weights and personalized classes with fixed weights, achieved WER reduction of 12%. This gain was largely preserved even after interpolation with a strong baseline. We plan to continue our investigations of WPE language models focusing on scaling and supporting more underlying LM types such as continuous space LMs.

7. References

[1] Kuhn, R. and De Mori, R.: "A Cache-based Natural Language Model for Speech Recognition"; in IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(6), pp.570-583, 1990.

[2] Lau, R., Rosenfeld, R. and Roukos S.: "Trigger-based Language Models: A Maximum Entropy Approach"; in Proc. of ICASSP, 1993.

[3] Haidar, M.A. and O'Shaughnessy, D.: "Topic n-gram Count Language Model Adaptation for Speech Recognition; in Proc. of SLT, pp.165-169, 2012.

[4] Mikolov, T. and Zweig G: "Context Dependent Recurrent Neural Network Language Model"; in Proc. of SLT, pp.234-239, 2012.

[5] Brown, P. F., deSouza, P. V., Mercer, R. L., Della Pietra, V. J. and Lai, J. C.: "Class-based n-gram Models of Natural Language"; in Comput. Linguistics, 18(4), pp.467–479, 1992.

[6] Kuo, H. K. J. and Reichl, W.: "Phrase-based Language Models for Speech Recognition"; in proc. of Eurospeech, 1999.

[7] Saon, G., Padmanabhan, M.: "Data-driven Approach to Designing Compound Words for Continuous Speech Recognition"; in IEEE trans on Speech and Audio Processing, 9(4), 2001, pp.327–332.

[8] Levit, M., Parthasarathy, S., Chang, S., Stolcke, A. and Dumoulin, B.: "Word-Phrase-Entity Language Models: Getting More Mileage out of N-grams"; in Proc. Interspeech, 2014.

[9] Stolcke, A.: "SRILM — an Extensible Language Modeling Toolkit"; in proc. of Interspeech, 2002.

[10] Luxenburger, J., Elbassuoni, S. and Weikum, G.: "Matching Task Profiles and User Needs in Personalized Web Search"; in Proc. of CIKM, pp.689698, ACM, 2008.

[11] Krestel, R. and Fankhauser, P.: "Language Models and Topic Models for Personalizing Tag Recommendation"; in Proc. of International Conference on Web Intelligence and Intelligent Agent Technology, vol.1, pp.82-89, 2010.

[12] Paek, T. and Chickering, D.: "Improving Command and Control Speech Recognition on Mobile Devices: Using Predictive User Models for Language Modeling"; in "User Modeling and User-Adapted Interaction", Special Issue on Statistical and Probabilistic Methods for User Modeling, 17(1-2), pp.93- 117, 2007.

[13] Wen, T. H., Heidel, A., Lee, H. Y., Tsao, Y. and Lee, L. S.: "Recurrent Neural Network Based Language Model Personalization by Social Network Crowdsourcing"; in Proc of Interspeech, pp.2703-2707, 2013.

[14] Zweig, G. and Chang, S.: "Personalizing Model M for Voice-Search"; in Proc. of Interspeech, pp.609-612, 2011.

[15] Slaney, M., Stolcke, A. and Hakkani-Tür, D.: "The Relation of Eye Gaze and Face Pose: Potential Impact on Speech Recognition"; in Proc. of 16th ACM International Conference on Multimodal Interaction, 2014.

[16] Anastasakos, T., Kim, Y.-B. and Deoras, A.: "Task Specific Continuous Word Representations for Mono and Multi-lingual Spoken Language Understanding"; in Proc. of ICASSP, 2014.

[17] Levit, M., Stolcke, A., Chang, S., Parthasarathy, S.: "Token-level Interpolation for Class-Based Language Models": in Proc. IEEE ICASSP, 2015.

[18] Liu, X., Gales, M. J. F. and Woodland, P. C.: "Context Dependent Language Model Adaptation"; in Proc. of Interspeech, 2008.