DOMAIN-AWARE NEURAL LANGUAGE MODELS FOR SPEECH RECOGNITION

Linda Liu   Yile Gu   Aditya Gourav   Ankur Gandhe

Shashank Kalmane   Denis Filimonov   Ariya Rastrow   Ivan Bulyko

Amazon Alexa

ABSTRACT

As voice assistants become more ubiquitous, they are increasingly expected to support and perform well on a wide variety of use-cases across different domains. We present a domain-aware rescoring framework suitable for achieving domain-adaptation during second-pass rescoring in production settings. In our framework, we fine-tune a domain-general neural language model on several domains, and use an LSTM-based domain classification model to select the appropriate domain-adapted model to use for second-pass rescoring. This domain-aware rescoring improves the word error rate by up to 2.4% and slot word error rate by up to 4.1% on three individual domains – shopping, navigation, and music – compared to domain general rescoring. These improvements are obtained while maintaining accuracy for the general use case.

Index Terms— language modeling, second-pass rescoring, domain adaptation, automatic speech recognition

1. INTRODUCTION

Voice assistants have become increasingly popular and are used for an ever-expanding set of use-cases. For example, users can currently ask voice assistants to play music (music domain), obtain a business address (navigation domain), and buy an item (shopping domain). Often, these automatic speech recognition systems consist of a separate language model (LM) component in the first-pass used in conjunction with an acoustic model, and an optional LM in the second-pass for rescoring. These LM models are trained to estimate the probability of a sequence of words $P(w_1..wn)$. While n-gram LMs do this by estimating the probability of each word, given the previous $n - 1$ words, neural LMs (NLMs) learn a distributed representation for words as well as the probability function for the word sequence in context [1]. This allows them to generalize estimates to unseen word sequences, and for longer word histories. Both types of LMs can be trained on a variety of textual data sources, and are typically optimized for the general use case.

However, different usage *domains* may differ significantly in their language statistics and a general LM may not perform as well on new domains, or domains otherwise not well represented by the general use case [2]. Training a single LM

that performs well on many domains is challenging. Some recent approaches include using attention to acoustic embedddings during decoding [3] or a compositional neural language model that learns how to combine multiple component language models [4].

In this work, we describe a domain-aware rescoring framework that can be used to address this in a production setting in the second-pass. We fine-tune a domain-general LM on data from three domains (music, navigation, shopping), and show that these models capture improvements in word error rate (WER) and slot WER (SlotWER) – which measures the performance on certain critical content words – beyond the general LM. We find this holds even when using neural models capable of capturing longer word histories and more complex relationships than traditional n-gram LMs. We demonstrate an effective way to combine a classification model to determine which domain-adapted model to use in second-pass rescoring. With this framework, we are able to obtain 0.7%-4.1% improvement on domain WER and SlotWER. The experimental results that we report use an experimental ASR system that does not reflect the performance of the current Alexa ASR system in production.

2. PREVIOUS WORK

Previous work on LM adaptation has shown that incorporating contextual or domain-adapted knowledge in the LM can improve the performance of ASR systems. One way to achieve this is by dynamically adjusting the weights of an interpolated n-gram LM, based on the preceding text [5]. The interpolated LM can consist of different LMs that are optimized for different dialog states or applications [6], topic/domain [5, 7], or decomposition of topic factors [8, 9]. These LMs can be trained separately, and then the appropriate mixture selected at runtime. Additionally, a cache component, which maintains a representation of recently occuring words, can be combined with an interpolated n-gram model to adapt the model to a target domain, based on the recent history [10].

More recently, neural adapation approaches have been used to adapt a LM to a target domain based on non-linguistic contextual signals, such as the application at the time of the request [11], or learned topic vectors [12, 13]. For example,

[12] used topic representations obtained from latent dirichlet allocation to adapt an NLM for genres and shows in a multi-genre broadcast transcription task. Domain-adaptation can also be achieved via shallow-fusion, in which an external (contextually constrained) LM is integrated during beam search [14].

Given the limited search space in the second-pass, we can use larger and more complex models in the second-pass of ASR systems, resulting in additional WER improvements [1]. In this approach, some set of hypotheses (represented as an n-best list, lattice, or confusion network) generated by a domain-general first-pass model is then rescored by a domain-adapted model. The LM scores obtained from the first pass model are updated using some combination of the first and second pass LM scores.

Recently, [15] demonstrated the feasibility of selecting adapted models to use in a production setting, using a pipeline with a topic classifier to select the appropriate domain-adapted LM. They show that this setup results in improvements beyond using the domain-general model alone. However, their second-pass LM is a 3-gram model. Recurrent neural LMs consistantly outperform traditional n-gram models, particularly on less frequent tail words, and are capable of capturing more complex relationship between words [1, 16], so it is unclear whether domain-adapted improvements might still hold when using a stronger LM.

In the next section, we present our domain-aware rescoring framework, in which we select a domain-aware NLM for second-pass based on the first pass ASR output.

3. DOMAIN-AWARE RESCORING FRAMEWORK

Our proposed domain-aware rescoring framework is illustrated in Fig. 1. After first-pass decoding, the one-best ASR output is fed into a classification model, in order to determine which second-pass rescoring model to use. The second-pass rescoring model is selected if the posteriors for a given class meet a selected threshold. The first and second-pass LM scores are interpolated based on an optimization for minimizing WER. In the following sections, we describe each of these components in turn: we first describe the first pass ASR system that we will use. Then we describe the classification model used to select the second-pass rescoring model, based on the first pass output. Finally, we describe the domain-adapted NLMs used in second-pass rescoring.

3.1. First-pass model

The first-pass LM is Kneser-Ney [17] smoothed n-gram LM trained on a variety of data, including in-domain data (transcribed and semi-supervised data from real user-agent interactions [1]), and out-of-domain data (e.g., corpora available on

---

[1]The user data is anonymized and consists of only a subset of data used for our production system
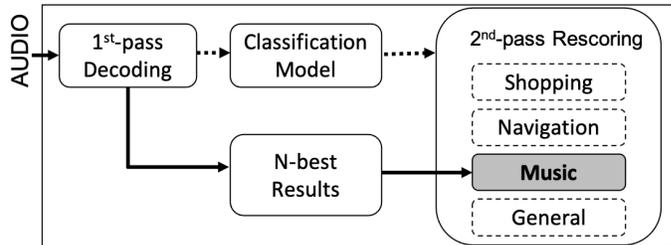

Fig. 1. Second-pass rescoring framework. The classification model select the appropriate rescoring model based on the first-pass decoding output. The n-best hypotheses are then rescored by the selected model (e.g., music-specific model).

the web). It is optimized to minimize perplexity over general traffic, which covers all domains [5]. The acoustic model is a low-frame-rate model with 2-layer frquency LSTM [18] followed by a 5-layer time LSTM trained with connectionist temporal classification loss [19].

3.2. Classification model

Selecting the appropriate second-pass rescoring model can be thought of as a text classification task. We train a light-weight LSTM-based classifier. It consists of an input embedding layer with 100 units and a vocabulary of roughly 150,000, including a token for out-of-vocabulary words, followed by an LSTM layer with 64 hidden units, a fully connected layer, and an output layer with 4 output units. Each output unit corresponds to one of the second-pass models (one domain-general model, three domain-adapted models). We use cross-entropy loss, softmax activation, as well as the Adam optimizer with intial learning rate of 0.001 and early stopping based on performance on a development set. At training time, the model is trained on annotated transcribed data. At test time, we feed in one-best ASR hypothesis. Input utterances were padded or truncated to a length of 10.

3.3. Domain-adapted NLMs

Following [20], we trained a NLM with two LSTM layers, each comprising 512 hidden units. We use noise contrastive estimatation based training, which is self-normalized and results in faster computation during training and inference, as it does not require the computation of the full softmax during training [21]. To adapt the NLMs to individual domains, we initialized training from the converged domain-general model, using the subset of data from each of the respective domains in order to fine-tune the LM to the domain [22]. Following [23], who used a smaller learning rate for fine-tuning, we used $1/4$ of the initial leaning rate, which yielded the best improvements on perplexity (PPL) from a range of $[1/8-1/2]$. In this way, each of our NLMs was trained on the same data, and also used the same vocabulary.

3.4. Second-pass rescoring and optimization

We generate a lattice from the 10-best ASR hypotheses, and rescore it using the second-pass model selected by the domain classifier. Rescoring itself uses the push-forward algorithm as described in [24, 25]. Additionally, we optimize the interpolation weights used to combine the first- and second-pass LMs based on (1) overall WER and (2) SlotWER. The optimization is carried out using the simulated annealing algorithm as described in [26].

We compare these results to an offline method where we use the one best ASR results to estimate per-utterance interpolation weights for each of our rescoring models using the EM algorithm. We then use these weights to do an linear interpolation between the models in the second-pass [5]. This approach is not practical in a production setting, but provides a point of comparison for our results.

4. EXPERIMENTS

4.1. Second-pass models

Model	PPL_{Other}	PPL_{Nav}	PPL_{Music}	PPL_{Shop}
LM_{Genrl}	0%	0%	0%	0%
LM_{Nav}	917.1%	**-47.7%**	1521.9%	314.5%
LM_{Music}	331.7%	734.3%	**-43.6%**	562.5%
LM_{Shop}	493.4%	349.6%	742.6%	**-45.8%**

Table 1. Relative PPL on evaluation set by domain split each domain-adapted LM compared to the domain-general LM baseline

The domain-general NLM was trained on 80 million utterances from anonymized user interactions with Alexa, constituting a portion of all live traffic. A portion of these utterances further included annotations for their domains. Among those utterances that were annotated, a portion were navigation, music, and shopping domain utterances (over 4 million each). We selected the top 238k most frequent words from this corpus as the vocabulary for the NLM, and mapped the out-of-vocabulary tokens to <unk>. The <unk> token was scaled by a factor of 10^{-5} in the rescoring experiments.

We adapted this domain-general NLM to three separate domains by fine-tuning of the annotated data for that domain: music, navigation, and shopping. Table 1 shows the perplexities for our test set, split out by domain, for each of the domain-adapted LMs and the domain-general LM. The *Other* partition does not contain music, navigation, shopping domain utterances, and only includes other domains, such as factual questions and commands to control connected home appliances. We observe a reduction in PPL on the evaluation data for the corresponding domain when using the domain-adapted model as compared to the domain-general model. However, we also observe an increase in the PPL of the *Other*

partition when using any domain-adapted model. This suggests that conditioning the second-pass LM on domain of the utterance can improve recognition, and supports the use of a domain classifier.

4.2. Domain classification model

Domain	Music	Navigation	Shopping	Other
Precision	0.94	0.92	0.92	0.97
Recall	0.93	0.89	0.97	0.92

Table 2. Per domain precision and recall for model classifier used to select the second-pass rescoring model

This model is trained with roughly 9 million utterances from user interactions with Alexa that were human-annotated for domain, using a 8-1-1 split for dev and eval (a subset of the total annotated data available to us). All utterances are first-turn interactions, in which the user directly prompts Alexa, and does not contain utterances that consist of only the wake-word (e.g., *"Alexa"*).

Across all domains, we observe an overall classification accuracy of 95.9%, with unweighted precision of 93.9% and recall of 92.8%, based on the max class. Per-domain classification breakdown is shown in Table 2. In our rescoring framework, we select the domain-adapted model for rescoring only in cases where the posteriors for that class meet some specified threshold (the higher the threshold, the more conservative the classification). Based on precision and recall values per domain, we used a 0.85 threshold across domains, resulting in an overall classification accuracy of 91%.

4.3. Evaluation Data

All of our experiments are evaluated on an evaluation set consisting of anonymized live user interactions with Alexa. The evaluation set consists of annotated data across several different Alexa devices for a select time period. The final evaluation set consists of roughly 10-20k utterances for each of the shopping, music, and navigation domains. The remainder of the set consists of 115k utterances from other domains. Additionally, for each domain, we annotate the popular contentful slots, in order to evaluate improvements in the SlotWER. These slots are the SongName, ArtistName, and AlbumName slots for Music, the PlaceName and StreetName slots for Navigation, and the ItemName slot for Shopping. Improving on recognition of the slot content is especially important for voice interactions, where the WER is higher. Slots are often specific to certain domains, making them most likely to benefit from domain-aware rescoring.

We created an development set using the same criteria as our evaluation set. This development set was used to determine the interpolation weights used to combine the LM scores from the first- and second-pass models.

Model	Navigation		Music		Shopping		Other
	WER	SlotWER	WER	SlotWER	WER	SlotWER	WER
LM_{Genrl}	-1.3%	-3.4%	-0.6%	1.0%	-2.2%	-0.3%	-0.8%
LM_{Nav}	-5.3%	-7.3%	–	–	–	–	1.8%
LM_{Music}	–	–	-2.6%	-4.1%	–	–	2.5%
LM_{Shop}	–	–	–	–	-4.0%	-4.1%	1.7%
DomainAware	-3.7%	-6.5%	-2.3%	-3.1%	-2.9%	-3.6%	-0.9%
AdaptationBaseline	-1.6%	-4.5%	-3.2%	-4.7%	-1.7%	-2.7%	-0.9%
Oracle	-31.2%	–	-25.2%	–	-32.8%	–	-18.4%

Table 3. WER improvements (shown as negative number) on domain-adapted test sets using general (LM_{Genrl}), domain-adapted LM ($LM_{Nav,Music,Shop}$), domain-aware rescoring with a domain classifier (DomainAware), or the dynamically interpolated model (AdaptationBaseline) as a second-pass rescorer. All relative comparisons are respect to the first-pass baseline (not shown). The Oracle provides a lower bound for the minimum WER achieveable, given a second pass model that always prefers the hypothesis in the nbest list with the lowest WER.

4.4. Results

We present WER results obtained using our domain-aware escoring framework (DomainAware), in which we rescore each utterance based on the model selected by the domain classifier (Table 3). We show these results split by each domain based on the annotation in our test set (Navigation, Music, Shopping), as well on the other domains (Other). The *Other* partition does not contain utterances from the aforementioned domains. Compared to the LM_{Genrl} alone, we find that domain-aware rescoring results show improvements of 0.7%-2.4% on WER and 3.1%-4.1% on SlotWER, across all domains. We also find a minor improvement (0.1% WERR) on the Other partition of the test set; these improvements are not obtained at the expense of general WER.

We also present the results that can be obtained on each evaluation set split, when rescoring that split using only the corresponding rescoring LM ($LM_{Nav,Music,Shop}$). This shows that the domain-adapted LMs can improve WER beyond the domain-general LM, even when trained using the same data. Compared to using LM_{Genrl}, we observe improvements of 1.8%-4.0% on WER and 3.8%-5.1% on SlotWER of the corresponding evaluation set split. Taken together, this shows that domain-aware rescoring allows us to capture the most of the improvements from using domain-adapted models, all while avoiding degradation on other domains.

We conducted the above comparisons using a weighted combination of the first and second pass models, with interpolation weights between first and second-pass models obtained by optimizing for WER or SlotWER on our development set. We observed no significant impact based on whether we used WER or SlotWER; consequently, all of the aforementioned results are obtained from WER optimization.

Finally, we compare the results obtained using domain-aware rescoring to the standard approach of dynamically adapting the interpolation weights over LM_{Genrl}, LM_{Nav}, LM_{Music}, and LM_{Shop}. This is the AdaptationBaseline shown in Table 3. From this, we see that the DomainAware approach is competitive with the dynamically interpolated adaptation baseline – both improve over the general neural language model, but DomainAware shows larger WER improvements for the navigation and shopping domains, and smaller WER improvements in the music domain. Though the DomainAware approach uses a domain classifier to select *one* model for rescoring, the single model used is trained to minimize perplexity on the respective domain; this is similar to the AdaptationBaseline, in which the interpolation weights are determined to minimize the perplexity for the word sequence, using a combination of all the models. However, unlike the DomainAware approach, the AdaptationBaseline approach is not practical for second-pass rescoring in production due to the latency incurred during weight optimization.

5. CONCLUSION

In this work, we present a framework for domain-aware rescoring that is suitable for use in a production environment. We show that using this framework enables us to capture WER improvements on domains beyond what is captured by a domain-general model trained on the same data. This framework has the benefit of allowing individual rescoring models to be maintained independently, making it suitable for asynchronous model updates. We demonstrate that domain-aware rescoring is competitive with traditional model interpolation methods, which can only be run in an offline environment when latency is not an issue. One additional benefit of a separate classification model is that individual domain-adapted models can be maintained and updated separately. Future work looks to combine the classification model and second-pass rescoring models into a single model; i.e., by incorporating an additional domain embeddding during the training of the NLM. The model would allow the NLM to dynamically adapt a specific domain, and can be further extended to other characteristics of the user or utterance beyond domain.

6. REFERENCES

[1] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[2] John Blitzer, *Domain Adaptation of Natural Language Processing Systems*, Ph.D. thesis, 2008.

[3] Ankur Gandhe and Ariya Rastrow, "Audio-attention discriminative language model for asr rescoring," in *ICASSP*. IEEE, 2020, pp. 7944–7948.

[4] Denis Filimonov, Ravi Teja Gadde, and Ariya Rastrow, "Neural composition: Learning to generate from multiple models," in *Self-Supervised Learning for Speech and Audio Processing Workshop*.

[5] Reinhard Kneser and Volker Steinbiss, "On the dynamic adaptation of stochastic language models," in *ICASSP*. IEEE, 1993, vol. 2, pp. 586–589.

[6] Frank Wessel, Andrea Baader, and Hermann Ney, "A comparison of dialogue-state dependent language models," in *ESCA Tutorial and Research Workshop (ETRW) on Interactive Dialogue in Multi-Modal Systems*, 1999.

[7] Anirudh Raju, Behnam Hedayatnia, Linda Liu, Ankur Gandhe, Chandra Khatri, et al., "Contextual language model adaptation for conversational agents," in *Interspeech*, 2018.

[8] Jerome R Bellegarda, "Exploiting latent semantic information in statistical language modeling," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1279–1296, 2000.

[9] Daniel Gildea and Thomas Hofmann, "Topic-based language models using em," in *Sixth European Conference on Speech Communication and Technology*, 1999.

[10] Roland Kuhn and Renato De Mori, "A cache-based natural language model for speech recognition," *TPAMI*, vol. 12, no. 6, pp. 570–583, 1990.

[11] Min Ma, Shankar Kumar, Fadi Biadsy, Michael Nirschl, Tomas Vykruta, and Pedro Moreno, "Modeling non-linguistic contextual signals in lstm language models via domain adaptation," in *ICASSP*. IEEE, 2018, pp. 6094–6098.

[12] Xie Chen, Tian Tan, Xunying Liu, Pierre Lanchantin, Moquan Wan, et al., "Recurrent neural network language model adaptation for multi-genre broadcast speech recognition," in *ISCA*, 2015.

[13] Tomas Mikolov and Geoffrey Zweig, "Context dependent recurrent neural network language model," in *2012 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2012, pp. 234–239.

[14] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 ICASSP*. IEEE, 2016, pp. 4945–4949.

[15] Dang Hoang Vu, Phuong Le-Hong, et al., "Online adaptation of language models for speech recognition," in *International Conference on Computer Recognition Systems*. Springer, 2019, pp. 164–173.

[16] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu, "Exploring the limits of language modeling," *arXiv:1602.02410*, 2016.

[17] Reinhard Kneser and Hermann Ney, "Improved backing-off for m-gram language modeling," in *ICASSP*. IEEE, 1995, vol. 1, pp. 181–184.

[18] Bo Li, Tara N Sainath, Arun Narayanan, Joe Caroselli, Michiel Bacchiani, et al., "Acoustic modeling for google home." in *Interspeech*, 2017, pp. 399–403.

[19] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006, pp. 369–376.

[20] Anirudh Raju, Denis Filimonov, Gautam Tiwari, Guitang Lan, and Ariya Rastrow, "Scalable multi corpora neural language models for asr," in *Interspeech*, 2019.

[21] Andriy Mnih and Yee Whye Teh, "A fast and simple algorithm for training neural probabilistic language models," *arXiv preprint arXiv:1206.6426*, 2012.

[22] Fadi Biadsy, Michael Alexander Nirschl, Min Ma, and Shankar Kumar, "Approaches for neural-network language model adaptation," in *Interspeech*, Stockholm, Sweden, 2017.

[23] Craig Atkinson, Brendan McCane, Lech Szymanski, and Anthony Robins, "Pseudo-recursal: Solving the catastrophic forgetting problem in deep neural networks," *arXiv preprint arXiv:1802.03875*, 2018.

[24] Michael Auli, Michel Galley, Chris Quirk, and Geoffrey Zweig, "Joint language and translation modeling with recurrent neural networks," in *EMNLP*, 2013, pp. 1044–1054.

[25] Shankar Kumar, Michael Nirschl, Daniel Holtmann-Rice, Hank Liao, Ananda Theertha Suresh, and Felix Yu, "Lattice rescoring strategies for long short term memory language models in speech recognition," in *2017 ASRU*. IEEE, 2017, pp. 165–172.

[26] Yang Xiang, Sylvain Gubian, Brian Suomela, and Julia Hoeng, "Generalized simulated annealing for efficient global optimization: the GenSA package for R.," *The R Journal Volume 5/1*, 2013.