Interspeech 2018
2-6 September 2018, Hyderabad, India

Contextual Language Model Adaptation for Conversational Agents

Anirudh Raju[1*], *Behnam Hedayatnia*[1*], *Linda Liu*[2], *Ankur Gandhe*[1], *Chandra Khatri*[1], *Angeliki Metallinou*[1], *Anu Venkatesh*[1], *Ariya Rastrow*[1]

[1]Amazon Alexa Machine Learning
[2] University of Rochester
[*]Both authors contributed equally to this work

{ranirudh,behnam,aggandhe,ckhatri,ametalli,anuvenk,arastrow}@amazon.com

Abstract

Statistical language models (LM) play a key role in Automatic Speech Recognition (ASR) systems used by conversational agents. These ASR systems should provide a high accuracy under a variety of speaking styles, domains, vocabulary and argots. In this paper, we present a DNN-based method to adapt the LM to each user-agent interaction based on generalized contextual information, by predicting an optimal, context-dependent set of LM interpolation weights. We show that this framework for contextual adaptation provides accuracy improvements under different possible mixture LM partitions that are relevant for both (1) Goal-oriented conversational agents where it's natural to partition the data by the requested *application* and for (2) Non-goal oriented conversational agents where the data can be partitioned using *topic labels* that come from predictions of a topic classifier. We obtain a relative WER improvement of 3% with a 1-pass decoding strategy and 6% in a 2-pass decoding framework, over an unadapted model. We also show up to a 15% relative improvement in recognizing named entities which is of significant value for conversational ASR systems.

Index Terms: speech recognition, language modeling, deep learning, weighted finite state transducers

1. Introduction

Automatic Speech Recognition (ASR) systems are a key component in building conversational agents. The most common approach to building language models (LMs) for ASR systems is to learn n-gram models on large text corpora. These models are trained to predict the conditional word probabilities given the context of the previous $n-1$ words. Hence, they do not model longer-range dependencies that may vary between different subsets of the training data such as speaking style, vocabulary and topics of conversation [1]. In this paper, we explore the use of contextual information for adapting the LM to each user-agent interaction. Broadly speaking, any additional information that would help predict the user's speech in the current interaction can be considered as contextual information, including the history of user-agent interactions, meta-data information such as dialog state and time of day, user personalized information like music preferences.

A common approach for LM adaptation is to represent the LM as a mixture of multiple component LMs, where the interpolation weights can be adapted to a single global target domain or dynamically adapted based on contextual information. In order to partition the training data and build the component LMs, a dominant approach has been to use supervised labels in the training data. In [2–4], the *dialog state* was used to build multiple component LMs and in [1, 5], supervised topic labels were used to build topic-specific component LMs. Other approaches which do not use supervised labels to partition the training data, rely on the availability of in-domain data to adapt LMs by selecting training data that minimizes cross-entropy [6], use a combination of mixture-based and MAP-based models [7] or use constrained KL divergence between unigram distributions [8]. Alternatively, clustered models are created either based on latent semantic analysis [9], k-means clustering [10] or through topic clusters computed by Latent Dirichlet Allocation (LDA) [11]. To estimate the context weights adaptively, past approaches used the expectation-maximization algorithm to learn the interpolation weights for a target sentence [10, 12]. More recently, using topic vectors directly in an RNN-LM was introduced in [13].

Our main contribution is to present a framework for adapting n-gram based LMs under some generalized contextual input. In this paper, we use a Deep Neural Network (DNN) to estimate the interpolation weights for a mixture of n-gram LMs, but the approach can be extended to other adaptation techniques such as n-gram boosting [14]. Unlike the approach presented in [13], an n-gram based adaptation can be easily integrated with an online system. For goal-oriented conversational interactions we partition our component LMs based on the *application label*, while for non-goal oriented, free-form dialogs we build component LMs based on an estimated *topic label*. For both use cases, our proposed framework outperforms an un-adapted framework, and leads to up to 6% relative WER reduction. Furthermore, our proposed adaptation leads to significant reduction, up to 15% relative WER, in recognizing topic-related entities of interest, such as person and location names, that appear in conversational interactions.

The paper is organized as follows. Section 2 describes the interpolated LM and our strategies for partitioning the data into component LMs, while Section 2.2 describes the on-the-fly LM interpolation. Sections 4 and 5 describe the experimental setup, results and discussion. Finally, we conclude in Section 6.

2. Interpolated Language Model

The interpolated language model is represented as a mixture of C component LMs, where the probability of a word sequence is calculated by using an interpolation weight λ_k corresponding to component LM_k in the mixture:

$$P_{mix}(w_i|h_i) = \sum_{k=1}^{C} \lambda_k P_k(w_i|h_i) \quad (1)$$

such that $\sum_{k=1} \lambda_k = 1$

2.1. Building component LMs

Building the component LMs requires splitting the LM training data into C components. The training data may be large transcribed corpora of user interactions or other external text cor-

10.21437/Interspeech.2018-1122

pora. Data should be partitioned along a meaningful dimension e.g., each component representing a different speaking style, vocabulary, discourse topic or some other nuance of language. In this work, we focus on ASR systems for two broad types of conversational agents, for which we use different data partitioning strategies.

Goal-oriented conversational agents: These agents are designed to support a few specific applications such as asking for weather information, playing music etc. Most interactions with personal assistants fall into this category. These interactions are typically short and comprise of very few dialog turns with the objective of providing the user with the requested information. Our goal-oriented interaction data is manually labeled in a straightforward manner based on the requested *application*, for example the sentence 'Play a song by the Beatles' would be labeled as 'Music' application. This enables splitting the training data based on the *application label*, and training one component LM for each application. We expect that user interactions with a music application would have different n-gram statistics compared to interactions with a shopping application.

Chatbots Chatbots are non goal-oriented conversational agents where the objective is to engage the user in an interesting and coherent conversation, as opposed to completing a specific task. Examples of older chatbots include ELIZA [15], while recent systems include the conversational agents that were deployed as part of the Alexa Prize competition [16]. Non goal-oriented human-chatbot dialogs typically contain a variety of conversation topics, therefore we choose the *topic label* as a natural way to partition the data for building the component LMs. Our data is not manually annotated with topic labels, so we use an off-the-shelf topic classifier designed for estimating conversational topics [17]. In future, this can be extended by unsupervised clustering of the training data.

2.2. Building component LMs on the fly

In weighted finite state transducers (WFST) based ASR systems [18], n-gram models are represented as WFSTs where each state represents the n-gram history h and the weight on an out-going arc is either the word probability $p(w_i|h)$ or the back-off weight αh. For each input utterance, an interpolated n-gram LM needs to be built based on interpolation weights. The proposed models for estimating the interpolation weights based on context are described in Section 3. After the weights are computed dynamically for each input utterance, we use an efficient on-the-fly interpolated WFST strategy [19] where the n-gram probabilities from each component LM are kept separate and interpolated only at run-time.

3. Estimation of interpolation weights

Given the interpolated LM representation as described in Equation 1, we need to estimate the interpolation weights λ_k during inference for each utterance based on the contextual information available. We propose to use a DNN model that estimates the λ_k's as the output of a softmax function, given any generic contextual feature input. The model can be trained in either a supervised or semi-supervised manner, as described below.

Minimize LM perplexity: The contextual adaptation model can be trained to estimate the interpolation weights λ_{ik} such that it maximizes the log-likelihood of each utterance in the training data under the interpolated model. This is equivalent to minimizing the LM perplexity of the training data. For an utterance $\boldsymbol{u_i} = (w_1, w_2....w_N)$ of N words, the perplexity-based loss function (PPL) is:

$$Loss = -logp(w_1, w_2...w_N) \quad (2)$$

$$= -\sum_N logp(w_j|w_{j-1}, ...w_{j-n-1}) \quad (3)$$

$$= -\sum_N log \sum_K \lambda_k p_k(w_j|w_{j-1}, ...w_{j-n-1}) \quad (4)$$

where p_k is the probability of the n-gram from LM_k. We compute the derivative of our loss function w.r.t. to λ_k and back-propagate the error to estimate the DNN parameters. To efficiently compute the loss, we pre-calculate $p_k(w_j|w_{j-1}, ...w_{j-n-1})$ for all examples as it stays constant throughout the optimization.

The training data can come from text corpus or from user-agent interactions. The user-agent interaction data can either be ASR 1-best recognitions (resulting in a semi-supervised training) or manually transcribed data (supervised training).

Minimize cross-entropy loss for component LM labels: The contextual adaptation model is trained to directly predict the intended component LM label for each utterance in the training data, using a cross-entropy (xent) loss function. The target labels come from manual annotations of the requested *application* for goal-oriented conversational agents, and from topic label estimates obtained from a topic classifier [17] for chatbots.

4. Experimental Setup

We build two separate ASR/LM systems, for goal-oriented and non goal-oriented conversations respectively. Section 4.1 describes the datasets used for training the LMs, while Section 4.2 provides details on LM building and the context adaptation model for each of the two cases.

4.1. Datasets

We have two datasets of user interactions with a conversational agent, specifically *goal-oriented* and *non goal-oriented* interactions, which are used for LM training and for evaluation. For our experiments, both datasets are split into the following partitions - 80% train, 10% dev, and 10% test. Additionally, we also have access to large external text corpora that are used only for LM training, and not as test sets.

Goal-oriented interaction data: This dataset consists of millions of utterances collected in far-field conditions from real user interactions with Alexa, a goal-oriented conversational agent. Each utterance corresponds to a single turn of user-agent interaction that has been annotated with the *application* that was requested and the corresponding *text transcription*. Application labels include Music, Shopping, Weather and others.

Non goal-oriented, chatbot interaction data: This dataset consists of hundreds of thousands of far-field speech-based user-agent interactions with a chatbot. The goal of the chatbot is to engage with the user in a conversation. The data is bucketed into *conversations*, where a single conversation is initiated and terminated by the user, and consists of multiple turns of user-agent interactions.

External text datasets We use a variety of external text corpora, from multiple sources such as news, voice-mail, web crawled corpora, etc. The total data size is of the order of billions words and it is entirely in the train partition.

4.2. ASR Systems

For all our experiments we use an experimental ASR system that does not reflect the performance of the production Alexa

Table 1: *Description of LM setup in each ASR system. Data sources are described in Section 4.1*

LM details	ASR System	
	Goal-oriented	**Chatbot**
Component LMs	Application	Topic
No. Comp. LMs	13	26
Training data	goal-oriented data	chatbot and external data

system. We build two ASR systems - one for goal-oriented agents and another for chatbots, each containing component LMs trained on different data, as described in Section 4.3. The structure of the LMs in both systems is a mixture of Katz smoothed [20] 4-gram language models which are interpolated on-the-fly in a WFST decoding framework, as described in Section 2.2. We also build baseline unadapted ASR systems for each of these use cases, by estimating static interpolation weights to minimize perplexity of the corresponding dev set.

4.3. Component LMs

In Table 1, for each ASR system, we describe the training data used to build the LMs, the number of component LMs, and what they represent (application vs topic label). LM training is described in Section 2.1. The goal-oriented ASR system uses the annotated application label to partition the LM training data, while the chatbot LM uses the topic labels estimated from a topic classifier [17]. Since the topic labels are obtained from a classifier as opposed to manual annotations, this scales easily and allows us to use large external text corpora in addition to user-agent interaction data, for training the chatbot LM. We use the following strategy to mix data from multiple external data sources for the chatbot LM - for each of the component topic based LMs, we build data source specific LMs which are statically interpolated to minimize perplexity on the corresponding dev set, i.e., dev partitions of the non goal-oriented chatbot conversations (Section 4.1)

4.4. Contextual adaptation model

The contextual adaptation model for each ASR system is trained from their respective user-agent conversational datasets (goal-oriented and non goal-oriented datasets of Section 4.1). We use a few hundred thousand utterances for training the contextual models. As described in Section 3, we train the models to either minimize cross-entropy (xent) of the component LM label distribution or perplexity (PPL) of the training data text. The DNN model shown in Figure 1 is a two layer network of 200 hidden units which is trained using the Adam optimizer, clipping gradients and early stopping. The λ_k parameters are estimated from the final softmax layer.

4.4.1. Contextual Features

The DNN-based contextual adaptation model allows for generic contextual features. The ones that we experiment with are obtained either from (1) a context window of past user-agent interactions or (2) the metadata information of the current interaction, i.e. time of day. Note that, these features are available prior to each user-agent interaction. The estimates from the DNN are used to build an adapted LM for each utterance, using which we run a 1-pass decoding with a real-time ASR system.

In addition, we report 2-pass decoding experiments where we use another feature - (3) 1-best hypothesis from the current interaction. The adapted LM estimated in this case is used to re-decode the utterance. This is suitable for a non-real time system,

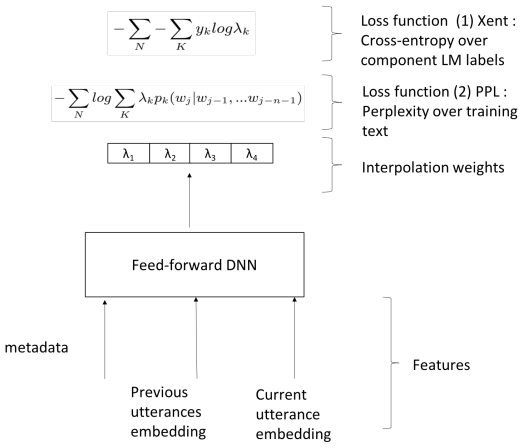

Figure 1: *Deep Neural Network (DNN) based contextual adaptation model to predict interpolation weights*

and provides an upper bound on the possible WER reduction, since the adapted LM is used as part of beam search decoding. Note that, in order to deploy this into a real-time system, we can run the first-pass decoding using the adapted LM from past utterance features and subsequently rescore the lattice using the more powerful adapted LM which includes current utterance features. This would result in a WER reduction in-between the 1-pass and 2-pass results reported here.

The context window that we used for the past utterance features depends on the type of conversation. For goal-oriented conversational agents, we use all user-agent interactions within the past T seconds of the current interaction. For non goal-oriented conversational agents, we use all user-agent interactions within the current conversation.

We have three top level features: past interactions based features (shorthand : *prev*), metadata based features (shorthand : *meta*) and current interaction based features (shorthand : *cur*). *prev* features include averaged pre-trained word embeddings [21] of all previous *user* turns within the context window and averaged word embeddings of all previous *agent* responses within the context window. *meta* features include the day of week and the time of day (morning, afternoon, evening). Lastly, *cur* features include the averaged word embeddings from 1-best ASR recognition of the current utterance.

5. Results and Discussion

Table 2: *Perplexity (PPL) and relative WER reduction (WERR) for the goal-oriented ASR system on a goal-oriented test set, for different objective functions and features for the DNN context model*

Model	Feats	PPL	WERR(%)	Entity WERR(%)
decoder : 1-pass				
No Adapt	-	30.66	-	-
DNN(Xent)	prev, meta	29.43	+0.33%	+1.17%
DNN(PPL)	prev, meta	26.10	-1.25%	+0.21%
decoder : 2-pass				
DNN(Xent)	prev, meta, cur	20.63	-3.49%	-3.04%
DNN(PPL)	prev, meta, cur	20.30	-3.24%	-2.21%

5.1. Overall results using both 1-pass and 2-pass decoding

Table 2 presents the perplexity and relative Word Error Rate (WER) reductions for the goal-oriented ASR system on a goal-oriented test set. We observe a relative WER reduction of 1.25% relative in a single-pass decoding framework and 3.5% relative in a 2-pass decoding framework by using the context adaptive

LM. Similarly, in Table 3, we present the results of the chatbot ASR system on a chatbot test set, where we see similar trends. Specifically, we observe a 2.7% rel. WER reduction in single-pass and a 6% rel. WER reduction in the 2-pass framework. The improved results for the 1-pass system are promising because they show that we can predict future behavior based on usage history and other metadata. As expected, the WER results are better in a 2-pass framework because knowledge of the current utterance helps us build a better adaptive LM. The current utterance features are the strongest signal to help improve WER and hence result in maximum improvements. However, this comes at the cost of higher latency of the 2-pass system compared to 1-pass.

5.2. Impact of different loss functions

From Tables 2 and 3 for both goal-oriented and chatbot ASR systems, we observe that when we train the context adaptation model using the perplexity (PPL) objective instead of cross-entropy (xent), we achieve adapted LMs with better PPL and WER on the test set. Moreover, training with the PPL objective function has the advantage of not requiring explicit labels of the optimal component LM for each utterance, and can be extended to scenarios where we may wish to train with semi-supervised data as described in Section 3.

Table 3: *Perplexity (PPL) and relative WER reduction (WERR) for the chatbot ASR system on a chatbot test set, for different objective functions and features for the DNN context model*

Model	**Feats**	**PPL**	**WERR(%)**	**Entity WERR(%)**
decoder : 1-pass				
No Adapt	-	60.77	-	-
DNN (Xent)	prev, meta	59.81	-1.73%	-8.19%
DNN (PPL)	prev, meta	58.14	-1.61%	-2.98%
DNN (PPL)	prev-d, meta	55.66	-2.76%	-10.92%
decoder: 2-pass				
DNN (PPL)	prev, cur, meta	42.03	-5.58%	-15.15%
DNN (PPL)	prev-d, cur, meta	42.83	-5.92%	-14.67%
DNN(PPL)	cur, meta	42.72	-5.98%	-15.32%
Topic model	cur	45.08	-5.52%	-13.14%

5.3. Impact of decaying past context

An interesting observation from our data is that users tend to interact with a chatbot in multiple turns, while goal-based interactions are significantly shorter. The typical context window for the chatbot system contains several more user-agent turns than for the goal-based system. Based on this insight, we further improved our contextual model by slightly modifying the contextual features used in the lengthy chatbot interactions. Specifically, we performed an exponentially decaying weighted average over the past word-embeddings, to give higher weight to recent utterances that are closer to the current utterance (called *prev-d* in Table 3). Using this exponential decay improves over using the standard *prev* features in both PPL and WER. In Table 3, using decaying context features (*prev-d*) leads to an overall WER reduction of 2.76% using the context adapted LM in a single-pass decoding framework and 5.92% in a 2-pass decoding framework.

5.4. Comparison with topic classifier based adaptation

We compare results when using the weights from the context adaptation model vs weights directly estimated from a topic classifier in a 2-pass decoding framework, see topic model vs DNN (PPL) in Table 3. For the topic model, the topic LM interpolation weights are the final output topic probabilities estimated by the topic classifier, which is the same classifier we used to partition the data into component LMs ([17], Section 4.3). From Table 3, we observe that using this strategy, leads to competitive results as compared to the context adaptation model that optimizes for PPL, i.e., leading to a 5.5% relative WER reduction compared to the chatbot baseline.

5.5. Impact on Named Entity accuracy

While WER is a standard metric to measure the performance of ASR systems, it does not capture the relative importance of different words in an utterance. For conversational systems, named entities such as people names, locations etc, are arguably more important compared to other words due to their impact on downstream tasks such as Natural Language Understanding (NLU). These entities tend to be topic specific, e.g., conversations about music contain entities such as artist and song names. Here, we analyze the accuracy of our proposed contextual ASR system on named entities. To measure entity error rate, we tag each word in our test data using an in-house Named Entity Recognition (NER) tagger. The entity error rate is defined as $(num_substitutions + num_deletions)/num_reference - words$. Note that we do not include insertions due to difficulty in attributing whether an insertion error was caused by the entity or the other surrounding words.

In the goal-oriented system, we bias towards the next application that the user is likely to request such as Music or Weather. Hence, in Table 2, we see similar improvements in the carrier phrase recognition as well as the entities i.e. the overall WER reduction is similar to the entity WER reduction. In contrast, the non-goal oriented data is more challenging due to it's free-form conversational nature, and contains topic specific named entities. We see a larger reduction of 15.32% entity error rate here in Table 3 with the best adapted LM. For example, difficult entities like "czechoslovakia" and "abigail" are correctly recognized by the adapted LM. This is a promising result for conversational ASR systems, where successful recognition of named entities is critical for completing the user request or engaging the user in meaningful dialog about topics of interest.

6. Conclusions and Future Work

We described methods for improving the performance of a mixture of n-gram LMs with on-the-fly interpolation, using a contextual adaptation framework. We presented a DNN-based method to predict an optimal set of interpolation weights for each interaction, in an online fashion, from generalized contextual information. This model was evaluated for both goal-oriented conversational agents where we partitioned the data by the requested application and on non-goal oriented conversational agents where we partitioned the data using topic labels that come from predictions of a topic classifier. We achieved a relative WER reduction up to 3% in a 1-pass decoding strategy using past context only, and up to 6% relative in a 2-pass decoding strategy, over a statically interpolated baseline LM. Our method benefits named entities the most: we reduced entity error rate by up to 15.32% relative. Future work includes evaluating this contextual LM adaptation strategy using other relevant context, such as topics derived from unsupervised clustering of the conversation, multimodal information that is available to the conversational agent or information about current events.

7. References

[1] R. Kneser and V. Steinbiss, "On the dynamic adaptation of stochastic language models," in *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 1993, pp. 586–589.

[2] F. Wessel, A. Baader, and H. Ney, "A comparison of dialogue-state dependent language models," in *Proc. ESCA Workshop on Interactive Dialogue in Multi-Modal Systems*, Irsee, Germany, Jun. 1999, pp. 93–96.

[3] W. Xu and A. I. Rudnicky, "Language modeling for dialog system," in *INTERSPEECH*, 2000, pp. 118–121.

[4] K. Visweswariah and H. Printz, "Language models conditioned on dialog state." in *INTERSPEECH*, 2001, pp. 251–254.

[5] J. Gao, H. Suzuki, and W. Yuan, "A comparative study on language model adaptation using new evaluation metrics," vol. 5/3. Association for Computational Linguistics, April 2006, pp. 207–227. [Online]. Available: https://www.microsoft.com/en-us/research/publication/a-comparative-study-on-language-model-adaptation-using-new-evaluation-metrics/

[6] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 355–362.

[7] L. Chen, J.-L. Gauvain, L. F. Lamel, G. Adda, and M. Adda, "Language model adaptation for broadcast news transcription," 2001.

[8] R. Kneser, J. Peters, and D. Klakow, "Language model adaptation using dynamic marginals." in *EUROSPEECH*, 1997.

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

[10] P. Clarkson and A. J. Robinson, "Language model adaptation using mixtures and an exponentially decaying cache," in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 1997, pp. 799–802.

[11] Y.-C. Tam and T. Schultz, "Dynamic language model adaptation using variational bayes inference," in *INTERSPEECH*, vol. 0, no. 0, 2005, pp. 5–8.

[12] K. Thadani, F. Biadsy, and D. M. Bikel, "On-the-fly topic adaptation for youtube video transcription," in *INTERSPEECH*, 2012, pp. 210–213.

[13] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model," in *2012 IEEE Spoken Language Technology Workshop (SLT)*, 2012.

[14] P. Aleksic, M. Ghodsi, A. Michaely, C. Allauzen, K. Hall, B. Roark, D. Rybach, and P. Moreno, "Bringing contextual information to google speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[15] J. Weizenbaum, "Eliza a computer program for the study of natural language communication between man and machine," *Communications of The ACM*, vol. 9, no. 1, pp. 36–45, 1966.

[16] The Alexa Prize Competition, "https://developer.amazon.com/alexaprize."

[17] F. Guo, A. Metallinou, C. Khatri, A. Raju, A. Venkatesh, and A. Ram, "Topic-based evaluation for conversational bots," *NIPS Workshop on Conversational AI*, 2017.

[18] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech and Language*, vol. 16, no. 1, pp. 69–88, 1 2002.

[19] B. Ballinger, C. Allauzen, A. Gruenstein, and J. Schalkwyk, "On-demand language model interpolation for mobile speech input," in *Interspeech*, 2010, pp. 1812–1815.

[20] S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," in *IEEE Transactions on Acoustics, Speech and Singal processing*, vol. ASSP-35, no. 3, March 1987, pp. 400–401.

[21] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162