Towards Improving Predictive AAC using Crowdsourced Dialogues and Partner Context

Keith Vertanen
Michigan Technological University
Houghton, MI, USA
vertanen@mtu.edu

ABSTRACT

Augmentative and Alternative Communication (AAC) devices typically rely on a language model to help make predictions or disambiguate user input. We investigate how to improve predictions in two-sided conversational dialogues. We collect and share a new corpus of crowdsourced everyday dialogues. We show how language models based on recurrent neural networks outperform N-gram models on these dialogues. We demonstrate further gains are possible using text obtained from an AAC user's communication partner, even when that text is partial or contains errors.

1. INTRODUCTION

Communicating with an AAC device poses many challenges such as generating text sufficiently fast to take part in conversations, controlling the delivery of pre-entered text, conversation turn-taking, and expressing emotion [3]. Here we focus on the problem of generating text quickly by investigating ways to improve AAC device predictions during real-time conversations.

Predictive AAC devices typically use a language model. The language style and two-sided nature of everyday conversations is different from many of the common data sources used to train language models.We show how crowd workers can easily generate everyday conversational dialogues and how recurrent neural network language models (RNNLMs) [4] improve modeling of these crowdsourced dialogues.

AAC devices typically use only the AAC user's side of a conversation. We show gains are possible by modeling both sides of the dialogue. This might be possible by performing speech recognition on the person speaking with an AAC user. Speech context was used in the Converser AAC interface which recognized noun phrases from the speaking person and incorporated them into utterance templates [7].

Here we use the entire speech recognition result as context to a language model trained on dialogues. We show gains are robust even to substantial simulated recognition errors. Finally, we show how knowing even a single word of context from a partner's turn can improve predictions.

ASSETS '17 October 29-November 1, 2017, Baltimore, MD, USA
© 2017 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-4926-0/17/10.
DOI: https://doi.org/10.1145/3132525.3134814

A:	What's your favorite dessert?
B:	Molten Chocolate Lava cake with Raspberries
A:	Oh my God, that sounds grand!
B:	Yes, let's go get some!
A:	I've been dying to try that new cafe. Maybe it's on their menu?
B:	I'll pull it up online and find out.

Table 1: A dialogue created by Amazon workers.

2. EXPERIMENTS

To generate data for our experiments, we had Amazon Mechanical Turk workers think of a question they might ask someone. These questions were shown to other workers who decided if a question was plausible and, if so, to create a response. We continued this process until six turns were completed. Each turn extension took about two hours and cost \$20. We collected 1,419 six-turn dialogues. We used 60% as a training set, 20% as a development set, and 20% as a test set. Overall workers invented plausible and creative dialogues (Table 1). We have made the dialogues available[1].

2.1 Predicting AAC user turns

First, we investigated predicting the AAC user turns in the test set without using the partner dialogue turns. We assumed the 2nd, 4th, and 6th turns were the AAC user. We report per-world *perplexity* on these turns (8K total words). Perplexity measures the possible options for the next token given the current context, e.g. a random digit sequence 0-9 has a perplexity of 10. Lower perplexity is better.

We used all dialogue turns, treating each turn as an independent training example (46K total words). We trained interpolated modified Kneser-Ney 4-gram models with no count cutoffs and a 35K vocabulary. We used Twitter as a large training set as it is a close match to AAC-like data [6].

The dialogue data was word-for-word better than Twitter, lowering perplexity from 285 to 160 (Table 2). However, using 50M words of Twitter data was much better, cutting perplexity to 81. Instead of random Twitter data, we selected 50M words using cross-entropy difference selection [5] with an in-domain model trained on 25% or 100% of the dialogues. This lowered perplexity to 76 and 73 respectively. This shows the value of having a large number of dialogues.

We trained RNNLMs on the cross-entropy selected Twitter data. An RNNLM with 250 sigmoid units had a perplexity of 89 (Table 3). Using 250 gated recurrent units and Noise Contrastive Estimation (NCE) lowered perplexity to

[1] http://keithv.com/data/turk-dialogues.txt

Training data	Words	PPL
Twitter, small amount of data	46K	285
Crowd dialogues	46K	160
Twitter, large amount of data	50M	81
Twitter, select w/ 25% crowd dialogues	50M	76
Twitter, select w/ 100% crowd dialogues	50M	73

Table 2: N-gram perplexity varying training data.

Model	PPL
Baseline Twitter RNNLM	89
+ gated recurrent units, NCE	78
+ maximum entropy	67
+ interpolate Twitter 4-gram LM	63
+ unigram cache	62
+ dialogue RNNLM	61

Table 3: RNNLM perplexities with added features.

78. Training a maximum entropy model in the network lowered perplexity to 67. Next we linearly interpolated the RNNLM and the N-gram model with a weight optimized on the development set. This resulted in a perplexity of 63, better than either model alone. Since words often reoccur in a dialogue, we added a unigram cache which reduced perplexity to 62. Finally, adding an RNNLM trained on just the crowdsourced dialogues lowered perplexity to 61.

2.2 Using partner turns

We tested using context from the partner turns to help predict the AAC user turns. Due to our relatively small crowdsourced corpus, we instead used dialogues from movies [1]. We trained an RNNLM treating each dialogue as a training example (3.8M words, 81K dialogues). For comparison, we tested the model on just the AAC user turns without partner turns (the test set used previously). Without partner turns, the RNNLM had a perplexity of 93. If instead the model made predictions based on both sides of the dialogues, perplexity dropped to 81. Thus, the RNNLM was able to use partner turns to improve predictions. A 6-gram N-gram language model was less capable and had a perplexity of 95.

The previous experiments assumed a perfect transcript of partner turns. We simulated speech recognition errors by injecting an error at a word in a partner's turn with some probability. 80% of errors were substitution errors, 10% insertion, and 10% deletion. Substitution and insertion errors used a random in-vocabulary word. Even at error probabilities up to 0.7, partner turns improved predictions (Figure 1). We then interpolated the two-sided dialogue model with the best model from the previous section. With perfect partner transcripts, this lowered perplexity from 61 to 54. Using turns with a 0.2 error probability only slightly increased perplexity to 55. Thus it appears likely even with recognition errors, partner turns can inform predictions.

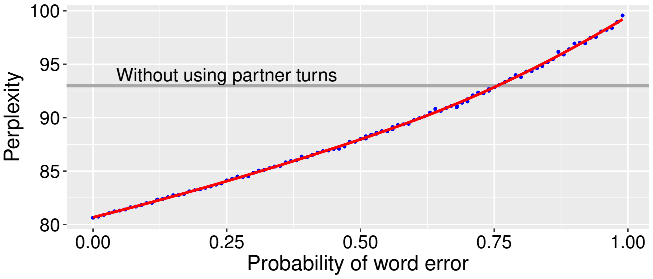

Figure 1: Perplexity with simulated rec. errors.

Instead of using the entire speech recognition transcript of a partner's turn, we could instead use just selected words, e.g. the non-stop words recognized with the highest confidence. Alternatively, a partner could explicitly suggest relevant words via a mobile app as in the AACrobat prototype [2]. We simulated this by using the word in the first turn with the lowest unigram probability. We trained an RNNLM on 1M words of Twitter data containing this word. As a baseline, we trained an RNNLM on 1M words of random Twitter data. The baseline RNNLM was interpolated with the two-sided RNNLM with partner turns having a 0.2 error probability. This reduced perplexity from 55 to 54. If instead we interpolated an RNNLM trained on data with the rarest word from the first turn, perplexity was 51.

3. CONCLUSIONS

Our experiments highlight some of the state-of-the-art language modeling methods available for improving predictive AAC. They also show how crowdsourcing dialogues can provide useful training data. We show how context obtained from speech recognition might be incorporated to improve AAC interface predictions. The speech-based improvements appear to be robust to recognition errors. This suggests speech context may be viable in real-world AAC interfaces.

We used only a small dialogue collection and a fraction of available Twitter data. Further work is needed to scale to larger amounts of data. We measured performance on dialogue turns contributed by crowd workers. Our improvements need to be validated in offline experiments with data from AAC users or in text entry studies with AAC users.

4. REFERENCES

[1] C. Danescu-Niculescu-Mizil and L. Lee. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proc. CMCL*, 2011.

[2] A. Fiannaca, A. Paradiso, M. Shah, and M. R. Morris. AACrobat: Using mobile devices to lower communication barriers and provide autonomy with gaze-based AAC. In *Proc. CSCW*, 2017.

[3] S. K. Kane, M. R. Morris, A. Paradiso, and J. Campbell. “At times avuncular and cantankerous, with the reflexes of a mongoose”: Understanding self-expression through augmentative and alternative communication devices. In *Proc. CSCW*, 2017.

[4] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Proc. Interspeech*, 2010.

[5] R. C. Moore and W. Lewis. Intelligent selection of language model training data. In *Proc. ACL*, 2010.

[6] K. Vertanen and P. O. Kristensson. The imagination of crowds: Conversational AAC language modeling using crowdsourcing and large data sources. In *Proc. EMNLP*, 2011.

[7] B. Wisenburn and D. J. Higginbotham. An AAC application using speaking partner speech recognition to automatically produce contextually relevant utterances: Objective results. *Augmentative and Alternative Communication*, 24(2).