The Imagination of Crowds: Conversational AAC Language Modeling using Crowdsourcing and Large Data Sources

Keith Vertanen
Department of Computer Science
Princeton University
vertanen@princeton.edu

Per Ola Kristensson
School of Computer Science
University of St Andrews
pok@st-andrews.ac.uk

Abstract

Augmented and alternative communication (AAC) devices enable users with certain communication disabilities to participate in everyday conversations. Such devices often rely on statistical language models to improve text entry by offering word predictions. These predictions can be improved if the language model is trained on data that closely reflects the style of the users' intended communications. Unfortunately, there is no large dataset consisting of genuine AAC messages. In this paper we demonstrate how we can crowdsource the creation of a large set of fictional AAC messages. We show that these messages model conversational AAC better than the currently used datasets based on telephone conversations or newswire text. We leverage our crowdsourced messages to intelligently select sentences from much larger sets of Twitter, blog and Usenet data. Compared to a model trained only on telephone transcripts, our best performing model reduced perplexity on three test sets of AAC-like communications by 60–82% relative. This translated to a potential keystroke savings in a predictive keyboard interface of 5–11%.

1 Introduction

Users with certain communication disabilities rely on augmented and alternative communication (AAC) devices to take part in everyday conversations. Often these devices consist of a predictive text input method coupled with text-to-speech output. Unfortunately, the text entry rates provided by AAC devices are typically low, between 0.5 and 16 words-per-minute (Trnka et al., 2009).

As a consequence, researchers have made numerous efforts to increase AAC text entry rates by employing a variety of improved language modeling techniques. Examples of approaches include adapting the language model to recently used words (Wandmacher et al., 2008; Trnka, 2008), using syntactic information (Hunnicutt, 1989; Garay-Vitoria and González-Abascal, 1997), using semantic information (Wandmacher and Antoine, 2007; Li and Hirst, 2005), and modeling topics (Lesher and Rinkus, 2002; Trnka et al., 2006). For a recent survey, see Garay-Vitoria and Abascal (2006).

While such language model improvement techniques are undoubtedly helpful, certainly they can all benefit from starting with a long-span language model trained on large amounts of closely matched data. For AAC devices this means closely modeling everyday face-to-face communications. However, a long-standing problem in the field is the lack of good data sources that adequately model such AAC communications. Due to privacy-reasons and other ethical concerns, there is no large dataset consisting of genuine AAC messages. Therefore, previous research has used transcripts of telephone conversations or newswire text. However, these data sources are unlikely to be an ideal basis for AAC language models.

In this paper we show that it is possible to significantly improve conversational AAC language modeling by first crowdsourcing the creation of a fictional collection of AAC messages on the Amazon Mechanical Turk microtask market. Using a care-

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 700–711,
Edinburgh, Scotland, UK, July 27–31, 2011.

fully designed microtask we collected 5890 messages from 298 unique workers. As we will see, word-for-word these fictional AAC messages are better at predicting AAC test sets than a wide-range of other text sources. Further, we demonstrate that Twitter, blog and Usenet data outperform telephone transcripts or newswire text.

While our crowdsourced AAC data is better than other text sources, it is too small to train high-quality long-span language models. We therefore investigate how to use our crowdsourced collection to intelligently select AAC-like sentences from Twitter, blog and Usenet data. We compare a variety of different techniques for doing this intelligent selection. We find that the best selection technique is the recently proposed cross-entropy difference method (Moore and Lewis, 2010). Using this method, we build a compact and well-performing mixture model from the Twitter, blog and Usenet sentences most similar to our crowdsourced data.

We evaluate our mixture model on four different test sets. On the three most AAC-like test sets, we found substantial reductions in not only perplexity but also in potential keystroke savings when used in a predictive keyboard interface. Finally, to aid other AAC researchers, we have publicly released our crowdsourced AAC collection, word lists and best-performing language models[1].

2 Crowdsourcing AAC-like Messages

As we mentioned in the introduction, there are unfortunately no publicly available sources of genuine conversational AAC messages. We conjectured we could create surrogate data by asking workers on Amazon Mechanical Turk to imagine they were a user of an AAC device and having them invent things they might want to say. While crowdsourcing is commonly used for simple human computation tasks, such as labeling images and transcribing audio, it is an open research question whether we can leverage workers' creativity to invent plausible and useful AAC-like messages. In this section, we describe our carefully constructed microtask and compare how well our collected messages correspond to communications from actual AAC users.

[1] http://www.aactext.org/imagine/

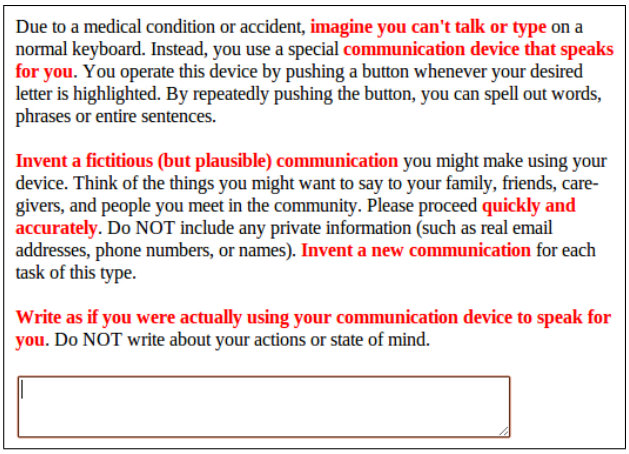


Figure 1: The interface for HITs of type 1 in our crowdsourced data collection.

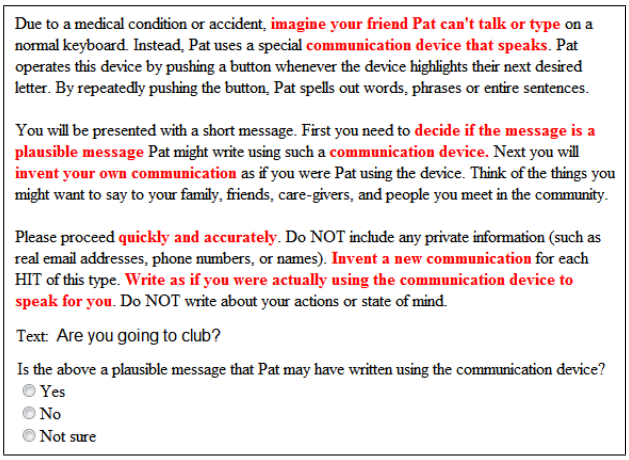


Figure 2: The interface for HITs of type 2 in our crowdsourced data collection.

2.1 Collection Tasks

To collect our data, we used two different types of human intelligence tasks (HITs). In type 1, the workers were told to imagine that due to an accident or medical condition they had to use a communication device to speak for them. Workers were asked to invent a plausible communication. Workers were prevented from pasting text. After several pilot experiments, we arrived at the instructions shown in figure 1.

In type 2, a worker first judged the plausibility of a communication written by a previous worker (figure 2). After judging, the worker was asked to "invent a completely new communication" as if the worker was the AAC user. Workers were prevented from pasting text or typing the identical text as the one just judged. The same communication

was judged by three separate workers. In this work we did not make use of these judgments.

2.2 Data Cleaning

While most workers produced plausible and often creative communications, some workers entered obvious garbage. These workers were identified by a quick visual scan of the submitted communications. We rejected the work of 9% of the workers in type 1 and 4% of the workers in type 2. After removing these workers, we had 2481 communications from type 1 and 4440 communications from type 2.

After combining the data from all accepted HITs, we conducted further semi-automatic data cleaning. We first manually reviewed communications sorted by worker. We removed workers whose text was non-fluent English or not plausible (e.g. some workers entered news headlines or proverbs). Identical communications from the same worker were removed. We removed communications with an out-of-vocabulary (OOV) rate of over 20% with respect to a large word list of 330K words obtained from human-edited dictionaries[2]. We also removed communications that were all in upper case, contained common texting abbreviations (e.g. "plz", "ru", "2day"), communications over 80 characters, and communications with excessive letter repetitions (e.g. "yippeeee"). After cleaning, we had 5890 messages from 298 unique workers.

2.3 Results

Tables 1 and 2 show some example communications obtained in each HIT type. Sometimes, but not always, type 2 resulted in the worker writing a similar communication as the one judged. This is a mixed blessing. While it may reduce the diversity of communications, we found that workers were more eager to accept HITs of type 2. The average HIT completion time was also shorter, 24 seconds in type 2 versus 36 seconds in type 1. While we initially paid $0.04/HIT for both types, we found in subsequent rounds that we could pay $0.02/HIT for type 2. We also had to reject less work in type 2 and qualitatively found the communications to be more AAC-like. Since workers had to imagine themselves in a very unfamiliar situation, it appears that providing a concrete example was helpful to workers.

[2]We combined Wiktionary, Webster's dictionary provided by Project Gutenberg, the CMU pronouncing dictionary and GNU aspell.

Is the dog friendly?
Can I have some water please?
I need to start making a shopping list soon.
What I would really like right now is a plate of fruit.
Who will drive me to the doctor's office tomorrow?

Table 1: Example communications from type 1.

Can you bring my slippers?
I am cold, is there another blanket.
How did Pam take the news?
Bring the fuzzy slippers here.
Did you have breakfast?
why are you so late?
I am pretty hungry, can we go eat?
I had bacon eggs and hashbrowns for breakfast.

Table 2: Example communications from type 2. The text in bold is the message workers judged. It is followed in plain text by the workers' new messages.

3 Comparison of Training Sources

In this section, we compare the predictive performance of language models trained on our Turk AAC data with models trained on other text sources. We use the following training sets:

- NEWS – Newspaper articles from the CSR-III (Graff et al., 1995) and Gigaword corpora (Graff, 2003). 60M sentences, 1323M words.
- WIKIPEDIA – Current articles and discussion threads from a snapshot of Wikipedia (January 3, 2008). 24M sentences, 452M words.
- USENET – Messages from a Usenet corpus (Shaoul and Westbury, 2009). 123M sentences, 1847M words.
- SWITCHBOARD – Transcripts of 2217 telephone conversations from the Switchboard corpus (Godfrey et al., 1992). Due to its conversational style, this corpus has been popular for AAC language modeling (Lesher and Rinkus, 2002; Trnka et al., 2009). 0.2M sentences, 2.6M words.
- BLOG – Blog posts from the ICWSM corpus (Burton et al., 2009). 25M sentences, 387M words.

- TWITTER – We collected Twitter messages via the streaming API between December 2010 and March 2011. We used the free Twitter stream which provides access to 5% of all tweets. Twitter may be particularly well suited for modeling AAC communications as tweets are short typed messages that are often informal person-to-person communications. Twitter has previously been proposed as a candidate for modeling conversations, see for example Ritter et al. (2010). 7M sentences, 55M words.
- TURKTRAIN – Communications from 80% of the workers in our crowdsourced collection. 4981 sentences, 24860 words.

WIKIPEDIA, USENET, BLOG and TWITTER all consisted of raw text that required significant filtering to eliminate garbage, spam, repeated messages, XML tags, non-English text, etc. Given the large amount of data available, our approach was to throw away any text that did not appear to be a sensible English sentence. For example, we eliminated any sentence having a large number of words not in our 330K word list.

3.1 Test Sets

We evaluated our models on the following test sets:

- COMM – Sentences written in response to hypothetical communication situations collected by Venkatagiri (1999). We removed nine sentences containing numbers. This set is used throughout the paper. 251 sentences, 1789 words.
- SPECIALISTS – Context specific phrases suggested by AAC specialists[3]. This set is used throughout the paper. 952 sentences, 3842 words.
- TURKDEV – Communications from 10% of the workers in our crowdsourced collection (disjoint from TURKTRAIN and TURKTEST). This set will be used for initial evaluations and also to tune our models. 551 sentences, 2916 words.
- TURKTEST – Communications from 10% of the workers in our crowdsourced collection (disjoint from TURKTRAIN and TURKDEV). This set is used only in the final evaluation section. 563 sentences, 2721 words.

[3] http://aac.unl.edu/vocabulary.html

Test set	Sentence
COMM	I love your new haircut.
COMM	How many children do you have?
SPECIALISTS	Are you sure you don't mind?
SPECIALISTS	I'll keep an eye on that for you
SWITCHTEST	yeah he's a good actor though
SWITCHTEST	what did she have like

Table 3: Examples from three of our test sets.

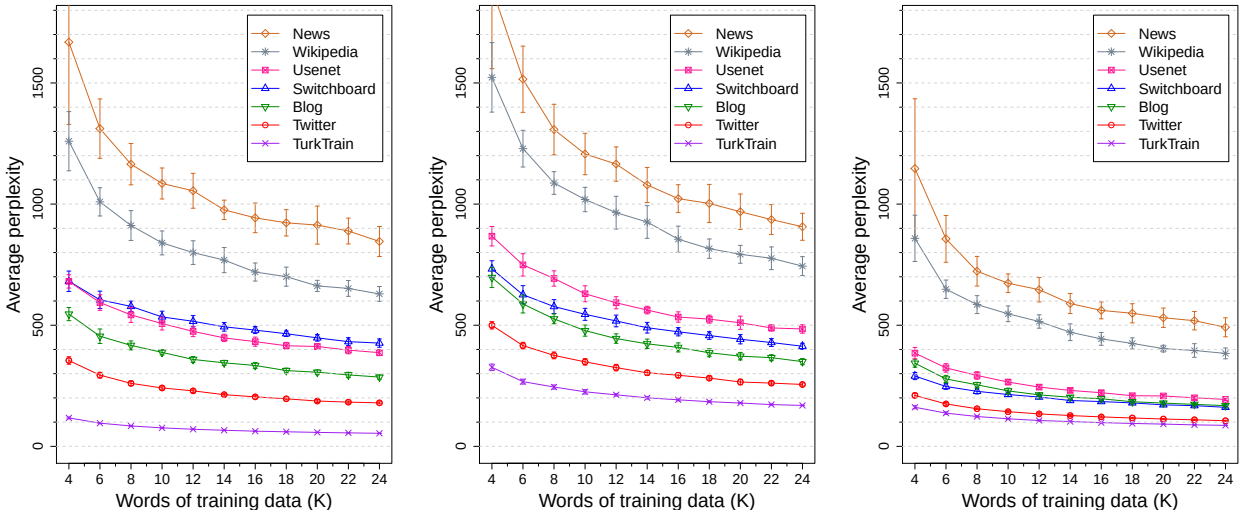
- SWITCHTEST – Transcripts of three Switchboard conversations (disjoint from the SWITCHBOARD training set). This is the same set used in Trnka et al. (2009). We dropped one sentence containing a dash. This set is only used in the final evaluation section. 59 sentences, 508 words.

TURKDEV and TURKTEST contain text similar to table 1 and 2. Table 3 shows some examples from the other three test sets. Sentences in COMM tended to be richer in vocabulary and subject matter than those in SPECIALISTS. The SPECIALISTS sentences tended to be general phrases that avoided mentioning specific situations, proper names, etc. Sentences in SWITCHTEST exhibited phenomena typical of human-to-human voice conversations (filler words, backchannels, interruptions, etc).

3.2 Language Model Training

All language models were trained using the SRILM toolkit (Stolcke, 2002). All models used interpolated modified Kneser-Ney smoothing (Kneser and Ney, 1995; Chen and Goodman, 1998). In this section, we trained 3-gram language models with no count-cutoffs. All text was converted to lowercase and we removed punctuation except for apostrophes. We believe punctuation would likely slow down a user's conversation for only a small potential advantage (e.g. improving text-to-speech prosody).

All models used a vocabulary of 63K words including an unknown word. We obtained our vocabulary by taking all words occurring in TURKTRAIN and all words occurring four or more times in the TWITTER training set. We restricted our vocabulary to words from our large list of 330K words. This restriction prevented the inclusion of common misspellings prevalent in many of our training sets. Our 63K vocabulary resulted in low OOV



(a) TURKDEV test set (b) COMM test set (c) SPECIALISTS test set

Figure 3: Perplexity of language models trained on the same amount of data from different sources. The perplexity is the average of 20 models trained on random subsets of the training data (one standard deviation error bars).

rates for all test sets: COMM 0%, SPECIALISTS 0.05%, TURKDEV 0.1%, TURKTEST 0.07%, and SWITCHTEST 0.8%.

3.3 Small Training Size Experiment

We trained language models on each dataset, varying the number of training words from 4K to 24K (the limit of the TURKTRAIN set). For each dataset and training amount, we built 20 different models by choosing sentences from the full training set at random. We computed the mean and standard deviation of the per-word perplexity of the set of 20 models.

As shown in figure 3, word-for-word the TURKTRAIN data was superior for our three most AAC-like test sets. Thus it appears our crowdsourcing procedure was successful at generating AAC-like data. TWITTER was consistently the second best. BLOG, USENET and SWITCHBOARD also performed well.

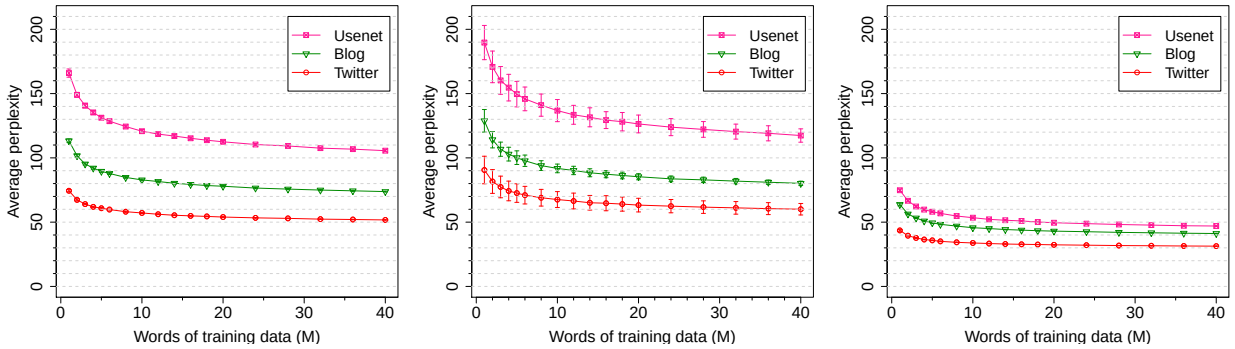
3.4 Large Training Size Experiment

The previous experiment used a small amount of training data. We selected the best three datasets having tens of millions of words of training data: USENET, BLOG, and TWITTER. As in the previous experiment, we computed the mean and standard deviation of the per-word perplexity of a set of 20 models. Increasing the amount of training data substantially reduced perplexity compared to our small TURKTRAIN collection (figure 4). Tweets were clearly well suited for modeling AAC-like text as 3M words of TWITTER data was better than 40M words of BLOG data.

3.5 Comparison with Real AAC Data

Beukelman et al. (1984) analyzed the communications made by five nonspeaking adults over 14 days. All users were experienced using a tape-typewriter AAC device. Beukelman gives a ranked list of the top 500 words, the frequency of the top 20 words, and statistics calculated on the communications.

For the top 10 words in Beukelman's AAC user data, we computed the probability of each word in our various datasets (figure 5). As shown, some words such as "to" and "a" occur with similar frequency across all datasets. Some words such as "the" are overrepresented in data such as news text. Other words such as "I" and "you" are much more variable. Our Turk data has the closest matching frequency for the most popular word "I". Interestingly, our Turk data shows a much higher probability for "you" than the AAC data. We believe this resulted from the situation we asked workers to imagine (i.e. communicating via a letter-at-a-time scanning interface). Workers presumed in such a situation they would need to ask others to do many tasks. We observed many requests in the data such as "Can



(a) TURKDEV test set (b) COMM test set (c) SPECIALISTS test set

Figure 4: Perplexity of language models trained on increasing amounts of data from three different training sources. Results on the TURKDEV, COMM and SPECIALISTS test sets.

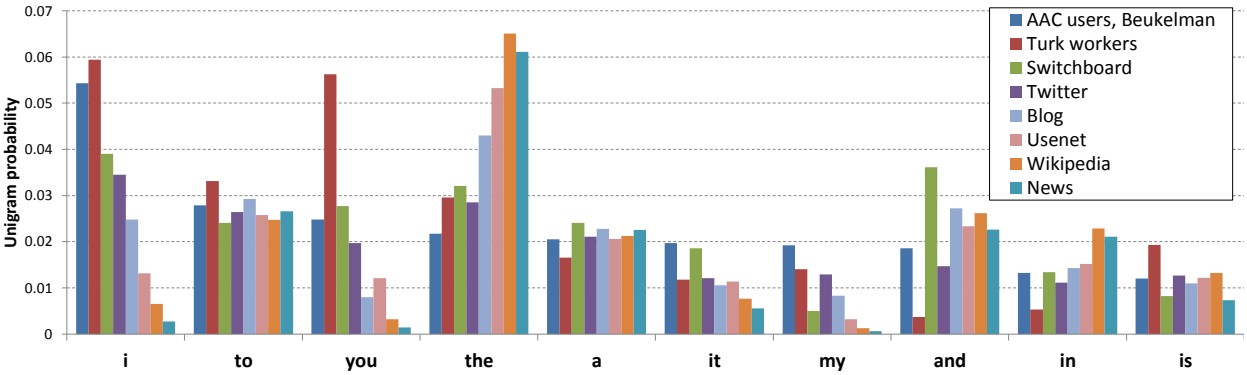


Figure 5: The unigram probabilities of the top 10 words reported by Beukelman et al. (1984).

you change my sheets?" and "Can you walk the dog for me?"

Beukelman reports 33% of all communications could be made using only the top 500 words. The same 500 words allowed writing of 34% of our Turk communications. Other datasets exhibited much lower percentages. Note that this is at least partially due to the longer sentences present in some datasets. Unfortunately, Beukelman does not report the average communication length. Our Turk communications were 5.0 words on average. The next shortest dataset was TWITTER with 7.5 words per communication. Despite their short average length, only 10% of Tweets could be written using the top 500 words.

Beukelman reports that 80% of words in the AAC users' communications were in the top 500 words. 81% of the words in our crowdsourced data were in this word list. For comparison, only 65% of words in our TWITTER data were in the 500 word vocabulary. While our TURKTRAIN set contains only 2141 unique words, this may in fact be good since it has been argued that rare words have received too much attention in AAC (Baker et al., 2000).

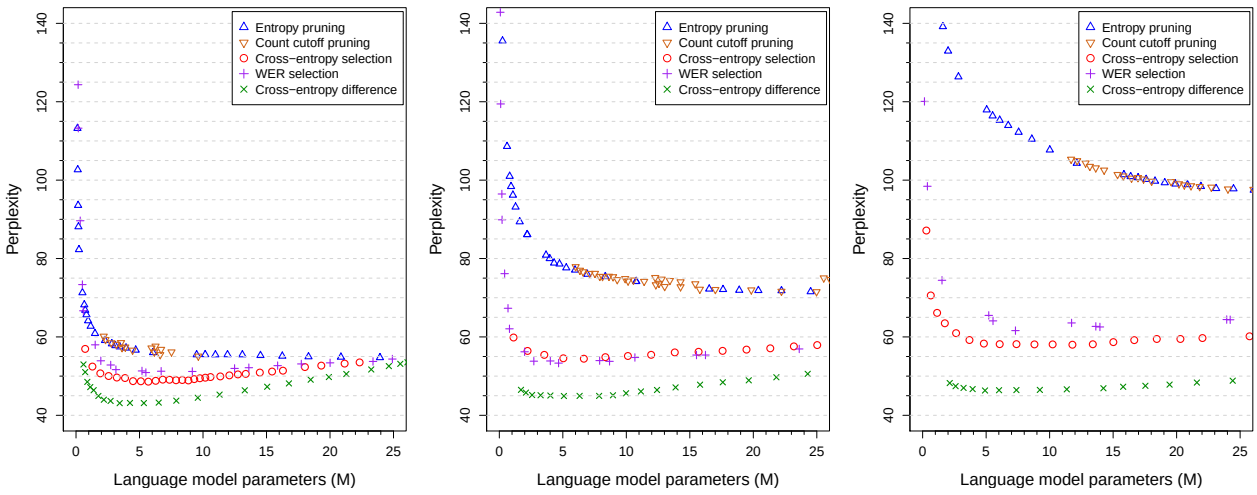
4 Using Large Datasets Effectively

In the previous section, we found our crowdsourced data was good at predicting AAC-like test sets. However, in order to build a good long-span language model, we would require millions of such communications. Crowdsourcing such a large collection would be prohibitively expensive. Therefore, we instead investigated how to use our crowdsourced data to intelligently select AAC-like data from other large datasets. For large datasets, we used TWITTER, BLOG and USENET as they were both large and well-matched to AAC data.

4.1 Selecting AAC-like Data

For each training sentence, we calculated three values:

- WER – The minimum word error rate between the training sentence and one of the crowdsourced



(a) TWITTER (b) BLOG (c) USENET

Figure 6: Perplexity on TURKDEV using different data selection and pruning techniques.

communications. This is the minimum number of words that must be inserted, substituted or deleted to transform the training sentence into a TURKTRAIN sentence divided by the number of words in the TURKTRAIN sentence. For example, the training sentence "I didn't sleep well Monday night either" was given a WER of 0.33 because two word-changes transformed it into a message written by a worker: "I didn't sleep well last night".

- Cross-entropy, in-domain – The average per-word cross-entropy of the training sentence under a 3-gram model trained on TURKTRAIN.
- Cross-entropy, background – The average per-word cross-entropy of the training sentence under a 3-gram model trained on a random portion of the training set. The random portion was the same size as TURKTRAIN.

We used these values to limit training to only AAC-like sentences. We tried three different selection methods. In *WER selection*, only sentences below a threshold on the word error rate were kept in the training data. This tends to find variants of existing communications in our Turk collection.

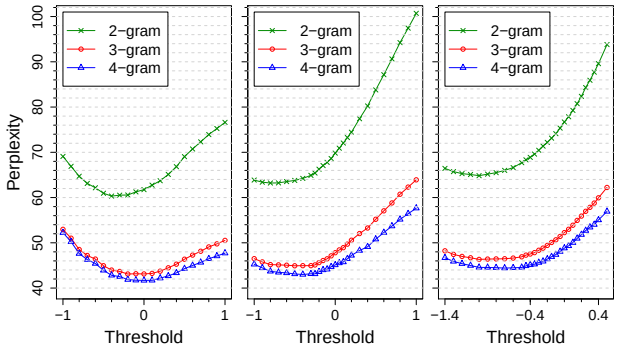
In *cross-entropy selection*, we used only sentences below a threshold on the per-word cross-entropy with respect to a TURKTRAIN language model. This is equivalent to placing a threshold on the perplexity. Previously this technique has been used to improve language models based on web data (Bulyko et al., 2007; Gao et al., 2002) and to construct domain-specific models (Lin et al., 1997).

In *cross-entropy difference selection*, a sentence's score is the in-domain cross-entropy minus the background cross-entropy (Moore and Lewis, 2010). This technique has been used to supplement European parliamentary text (48M words) with newswire data (3.4B words) (Moore and Lewis, 2010). We were curious how this technique would work given our much smaller in-domain set of 24K words.

4.2 Data Selection and Pruning

We built models selecting sentences below different thresholds on the WER, in-domain cross-entropy, or cross-entropy difference. For comparison, we also pruned our models using conventional count-cutoff and entropy pruning (Stolcke, 1998). During entropy pruning, we used a Good-Turing estimated model for computing the history marginals as the lower-order Kneser-Ney distributions are unsuitable for this purpose (Chelba et al., 2010).

We calculated the perplexity of each model on three test sets. We also tallied the number of model parameters (all n-gram probabilities plus all backoff weights). On TURKDEV, cross-entropy difference selection performed the best for all models sizes and for all training sets (figure 6). We also found cross-



(a) TWITTER (b) BLOG (c) USENET

Figure 7: Perplexity on TURKDEV varying the cross-entropy difference threshold.

entropy difference was the best on COMM, reducing perplexity by 10–20% relative compared to cross-entropy selection. Results on SPECIALISTS showed that WER and both forms of cross-entropy selection performed similarly. All three data selection methods were superior to count-cutoff or entropy pruning. We use cross-entropy difference selection for the remainder of this paper.

4.3 Model Order and Optimal Thresholds

We created 2-gram, 3-gram and 4-gram models on TWITTER, BLOG, and USENET using a range of cross-entropy difference thresholds. 4-gram models slightly outperformed 3-gram models (figure 7). The optimal threshold for 4-gram models were as follows: TWITTER 0.0, BLOG -0.4, and USENET -0.7. These thresholds resulted in using 20% of TWITTER, 5% of BLOG, and 1% of USENET.

4.4 Mixture Model

We created a mixture model using linear interpolation from the TWITTER, USENET and BLOG 4-gram models created with each set's optimal threshold. The mixture weights were optimized with respect to TURKDEV using SRILM. The final mixture weights were: TWITTER 0.42, BLOG 0.29, and USENET 0.29. Our final 4-gram mixture model had 43M total parameters and a compressed disk size of 316 MB.

5 Evaluation

In this section, we compare our mixture model against baseline models. We show performance with respect to usage in a typical AAC text entry interface based on word prediction.

5.1 Predictive Text Entry

Many AAC communication devices use word predictions. In a word prediction interface users type letters and the interface offers word completions based on the prefix of the current word and often the prior text. By selecting one of the predictions, the user can potentially save keystrokes as compared to typing out every letter of each word.

We assume a hypothetical predictive keyboard interface that displays five word predictions. Our keyboard makes predictions based on up to three words of prior context. Our keyboard predicts words even before the first letter of a new word is typed. As a user types letters, predictions are limited to words consistent with the typed letters. If the system makes a correct prediction, we assume it takes only one keystroke to enter the word and any following space.

We only predict words in our 63K word vocabulary (empty prediction slots are possible). We display a word even if it was already a proposed prediction for a shorter prefix of the current word. The first word in a sentence is conditioned on the sentence-start pseudo-word. If an out-of-vocabulary word is typed, the word is replaced in the language model's context with the unknown pseudo-word.

We evaluate our predictive keyboard using the common metric of keystroke savings (KS):

$$KS = \left(1 - \left(\frac{k_p}{k_a}\right)\right) \times 100\%,$$

where k_p is the number of keystrokes required with word predictions and k_a is the number of keystrokes required without word prediction.

5.2 Predictive Performance Experiment

We compared our mixture model using cross-entropy difference selection with three baseline models trained on all of TWITTER, SWITCHBOARD and TURKTRAIN. The baseline models were unpruned 4-gram models trained using interpolated modified Kneser-Ney smoothing. They had 72M, 5M, and 129K parameters respectively.

As shown in table 4, our mixture model performed the best on the three most AAC-like test sets (COMM, SPECIALISTS, and TURKTEST). The

LM	Test set	PPL	KS
Mixture	**COMM**	**47.9**	**62.5%**
Twitter	COMM	55.9	60.9%
Switchboard	COMM	151.1	54.4%
Turk	COMM	165.9	52.7%
Mixture	**SPECIALISTS**	**25.7**	**63.1%**
Twitter	SPECIALISTS	27.3	61.9%
Switchboard	SPECIALISTS	64.5	57.7%
Turk	SPECIALISTS	85.9	52.8%
Mixture	**TURKTEST**	**31.2**	**62.0%**
Twitter	TURKTEST	42.3	59.3%
Switchboard	TURKTEST	172.5	50.6%
Turk	TURKTEST	51.0	57.6%
Mixture	SWITCHTEST	174.3	52.8%
Twitter	SWITCHTEST	142.6	54.9%
Switchboard	**SWITCHTEST**	**79.2**	**58.8%**
Turk	SWITCHTEST	642.5	42.9%

Table 4: Perplexity (PPL) and keystroke savings (KS) of different language models on four test sets. The bold line shows the best performing language model on each test set.

mixture model provided substantial increases in keystroke savings compared to a model trained solely on Switchboard. The mixture model also performed better than simply training a model on a large amount of Twitter data. The model trained on only 24K words of Turk data did surprisingly well given its extremely limited training data.

Our Switchboard model performed the best on SWITCHTEST with a keystroke savings of 58.8%. For comparison, past work reported a keystroke savings of 55.7% on SWITCHTEST using a 3-gram model trained on Switchboard (Trnka et al., 2009). While our mixture model performed less well on SWITCHTEST (52.8%), it is likely the other three test sets better represent AAC communications.

5.3 Larger Mixture Model Experiment

Our mixture language model used the best thresholds with respect to TURKDEV. This resulted in throwing away most of the training data. This might be suboptimal in practice if an AAC user's communications are somewhat different or more diverse than the language generated by the Turk workers.

We trained a series of mixture models in which we varied the cross-entropy difference thresholds

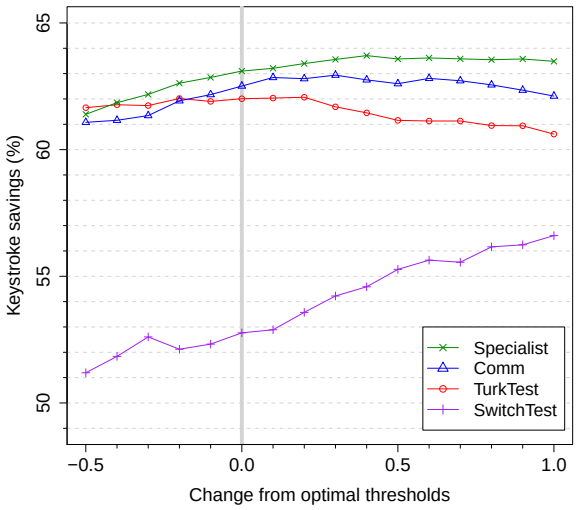


Figure 8: Keystroke savings on mixture models varying a constant added to the optimal thresholds with respect to TURKDEV.

by adding a constant to all three thresholds. The mixture weights for each new model were optimized with respect to TURKDEV. Using somewhat larger models did improve keystroke savings for all test sets except for TURKTEST (figure 8). However, using too large thresholds eventually hurt performance except on SWITCHTEST. Performance on SWITCHTEST steadily increased from 52.8% to 56.6%. These gains however came at the cost of bigger models. The model using +1.0 of the optimal thresholds had 384M parameters and a compressed size of 3.0 GB.

6 Discussion

Given the ethical implications of collecting messages from actual AAC users, it is unlikely that a large corpus of genuine AAC messages will ever be available to researchers. An important finding in this paper is that crowdsourcing can be an effective way to obtain surrogate data for improving AAC language models. Another finding is that Twitter provides a continuous stream of large amounts of very AAC-like data. Twitter also has the advantage of allowing models to be continually updated to reflect current events, new vocabulary, etc.

6.1 Limitations and Implications

We collected data from a large number of workers, some of whom may have written only a single com-

munication. This may have resulted in more messages about simple situations and perceived needs which could differ from true AAC usage.

Our data does not contain long-term two-sided conversations. Thus it may not be as useful for evaluating techniques that adapt to past messages or that use the conversation partner's communications.

We asked workers to imagine they were using a scanning-style AAC device. We believe this led workers to presume they would require assistance in many routine physical tasks. Our workers were (presumably) without cognitive or language impairments. Thus our collection is more representative of one subgroup of AAC communicators (scanning users with normal cognitive function and language skills). By modifying the situation given to workers, it is likely we can expand our collection to better represent other groups of AAC users, such as those using predictive keyboards or eye-trackers. However, obtaining data representative of users with cognitive or language impairments via crowdsourcing would probably be difficult.

While we were unable to obtain real AAC messages for testing, we believe the COMM and SPECIALIST test sets provide a good indication of the real-world potential for our methods. Our collected Turk data was compared with reported data from actual AAC users (though this comparison was necessarily coarse-grained). We hope that by releasing our data and models it may be possible for those privy to real AAC communications to validate and report about the techniques described in this paper.

We evaluated our models in terms of perplexity and keystrokes savings within the auspices of a predictive keyboard. Further work is needed to investigate how our numeric gains translate to real-world benefits to users. However, past work indicates more accurate predictions do in fact yield improvements in human performance (Trnka et al., 2009).

Finally, while the predictive keyboard is a commonly studied interface, it is not appropriate for all AAC users. Eye-tracker users may prefer an interface such as Dasher (Ward and MacKay, 2002). Single-switch users may prefer an interface such as Nomon (Broderick and MacKay, 2009). Any AAC interface based on word- or letter-based predictions stands to benefit from the methods described in this paper.

7 Conclusions

In this paper we have shown how workers' creativity on a microtask crowdsourcing market can be used to create fictional but plausible AAC communications. We have demonstrated that these messages model conversational AAC better than the currently used datasets based on telephone conversations or newswire text. We used our new crowdsourced dataset to intelligently select sentences from Twitter, blog and Usenet data.

We compared a variety of different techniques for intelligent training data selection. We found that even for our small amount of in-domain data, the recently proposed cross-entropy difference method was consistently the best (Moore and Lewis, 2010). Finally, compared to a model trained only on Switchboard, our best performing model reduced perplexity by 60-82% relative on three AAC-like test sets. This translated to a potential keystroke savings in a predictive keyboard interface of 5–11%.

In conclusion, we have shown how to create long-span AAC language models using openly available resources. Our models significantly outperform models trained on the commonly used data sources of telephone transcripts and newswire text. To aid other researchers, we have publicly released our crowdsourced AAC collection, word lists and best-performing models. We hope complementary techniques such as topic modeling and language model adaptation will provide additive gains to those obtained by training models on large amounts of AAC-like data. We plan to use our models to design and test new interfaces that enable faster communication for AAC users.

Acknowledgments

We thank Keith Trnka and Horabail Venkatagiri for their assistance. This work was supported by the Engineering and Physical Sciences Research Council (grant number EP/H027408/1).

References

Bruce Baker, Katya Hill, and Richard Devylder. 2000. Core vocabulary is the same across environments. In *California State University at Northridge Conference*.

David R. Beukelman, Kathryn M. Yorkston, Miguel Poblete, and Carlos Naranjo. 1984. Frequency of

word occurrence in communication samples produced by adult communication aid users. *Journal of Speech and Hearing Disorders*, 49:360–367.

Tamara Broderick and David J. C. MacKay. 2009. Fast and flexible selection with a single switch. *PLoS ONE*, 4(10):e7481.

Ivan Bulyko, Mari Ostendorf, Manhung Siu, Tim Ng, Andreas Stolcke, and Özgür Çetin. 2007. Web resources for language modeling in conversational speech recognition. *ACM Transactions on Speech and Language Processing*, 5(1):1–25.

Kevin Burton, Akshay Java, and Ian Soboroff. 2009. The ICWSM 2009 Spinn3r dataset. In *Proceedings of the 3rd Annual Conference on Weblogs and Social Media*.

Ciprian Chelba, Thorsten Brants, Will Neveitt, and Peng Xu. 2010. Study on interaction between entropy pruning and Kneser-Ney smoothing. In *Proceedings of the International Conference on Spoken Language Processing*, pages 2422–2425.

Stanley F. Chen and Joshua T. Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical report, Computer Science Group, Harvard University.

Jianfeng Gao, Joshua Goodman, Mingjing Li, and Kai-Fu Lee. 2002. Toward a unified approach to statistical language modeling for chinese. *ACM Transactions on Asian Language Information Processing*, 1:3–33.

Nestor Garay-Vitoria and Julio Abascal. 2006. Text prediction systems: A survey. *Universal Access in the Information Society*, 4:188–203.

Nestor Garay-Vitoria and Julio González-Abascal. 1997. Intelligent word-prediction to enhance text input rate. In *Proceedings of the 2nd ACM International Conference on Intelligent User Interfaces*, pages 241–244.

J.J. Godfrey, E.C. Holliman, and J. McDaniel. 1992. SWITCHBOARD: Telephone speech corpus for research and development. *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, pages 517–520.

David Graff, Roni Rosenfeld, and Doug Pau. 1995. CSR-III text. Linguistic Data Consortium, Philadelphia, PA, USA.

David Graff. 2003. English gigaword corpus. Linguistic Data Consortium, Philadelphia, PA, USA.

Sheri Hunnicutt. 1989. Using syntactic and semantic information in a word prediction aid. In *Proceedings of the 1st European Conference on Speech Communication and Technology*, pages 1191–1193.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, pages 181–184.

Gregory W. Lesher and Gerard J. Rinkus. 2002. Domain-specific word prediction for augmentative communication. In *Proceedings of the RESNA 2002 Annual Conference*.

Jianhua Li and Graeme Hirst. 2005. Semantic knowledge in word completion. In *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 121–128.

Sung-Chien Lin, Chi-Lung Tsai, Lee-Feng Chien, Ker-Jiann Chen, and Lin-Shan Lee. 1997. Chinese language model adaptation based on document classification and multiple domain-specific language models. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 1463–1466.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the 48th Annual Meeting of the Association of Computational Linguistics*, pages 220–224.

Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of twitter conversations. In *Proceedings of HLT-NAACL 2010*, pages 172–180.

Cyrus Shaoul and Chris Westbury. 2009. A USENET corpus (2005-2009). University of Alberta, Canada.

Andreas Stolcke. 1998. Entropy-based pruning of backoff language models. In *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*, pages 270–274.

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings of the 7th Annual International Conference on Spoken Language Processing*, pages 901–904.

Keith Trnka, Debra Yarrington, and Christopher Pennington. 2006. Topic modeling in fringe word prediction for AAC. In *Proceedings of the 11th ACM International Conference on Intelligent User Interfaces*, pages 276–278.

Keith Trnka, John McCaw, Debra Yarrington, Kathleen F. McCoy, and Christopher Pennington. 2009. User interaction with word prediction: The effects of prediction quality. *ACM Transactions on Accessible Computing*, 1:17:1–17:34.

Keith Trnka. 2008. Adaptive language modeling for word prediction. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Student Research Workshop*, pages 61–66.

Horabail Venkatagiri. 1999. Efficient keyboard layouts for sequential access in augmentative and alternative communication. *Augmentative and Alternative Communication*, 15(2):126–134.

Tonio Wandmacher and Jean-Yves Antoine. 2007. Methods to integrate a language model with semantic

information for a word prediction component. *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 506–513.

Tonio Wandmacher, Jean-Yves Antoine, Franck Poirier, and Jean-Paul Départe. 2008. SIBYLLE, an assistive communication system adapting to the context and its user. *ACM Transactions on Accessible Computing*, 1:6:1–6:30.

D. J. Ward and D. J. C. MacKay. 2002. Fast hands-free writing by gaze direction. *Nature*, 418(6900):838.