Discourse-Based Modeling for AAC

Margaret Mitchell **Richard Sproat**
Center for Spoken Language Understanding
Oregon Health & Science University
m.mitchell@abdn.ac.uk, rws@xoba.com

Abstract

This paper presents a method for an AAC system to predict a whole response given features of the previous utterance from the interlocutor. It uses a large corpus of scripted dialogs, computes a variety of lexical, syntactic and whole phrase features for the previous utterance, and predicts features that the response should have, using an entropy-based measure. We evaluate the system on a held-out portion of the corpus. We find that for about 3.5% of cases in the held-out corpus, we are able to predict a response, and among those, over half are either exact or at least reasonable substitutes for the actual response. We also present some results on keystroke savings. Finally we compare our approach to a state-of-the-art *chatbot*, and show (not surprisingly) that a system like ours, tuned for a particular style of conversation, outperforms one that is not.

Predicting possible responses automatically by mining a corpus of dialogues is a novel contribution to the literature on whole utterance-based methods in AAC. Also useful, we believe, is our estimate that about 3.5-4.0% of utterances in dialogs are in principle predictable given previous context.

1 Introduction

One of the overarching goals of Augmentative and Alternative Communication technology is to help impaired users communicate more quickly and more naturally. Over the past thirty years, solutions that attempt to reduce the amount of effort needed to input a sentence have include semantic compaction (Baker, 1990), and lexicon- or language-model-based word prediction (Darragh et al., 1990; Higginbotham, 1992; Li and Hirst, 2005; Trost et al., 2005; Trnka et al., 2006; Trnka et al., 2007; Wandmacher and Antoine, 2007), among others. In recent years, there has been an increased interest in whole utterance-based and discourse-based approaches (see Section 2). Such approaches have been argued to be beneficial in that they can speed up the conversation, thus making it appear more felicitous (McCoy et al., 2007). Most commercial tablets sold as AAC devices contain an inventory of canned phrases, comprising such items as common greetings, polite phrases, salutations and so forth. Users can also enter their own phrases, or indeed entire sequences of phrases (e.g., for a prepared talk).

The work presented here attempts to take whole phrase prediction one step further by automatically predicting appropriate responses to utterances by mining conversational text. In an actual deployment, one would present a limited number of predicted phrases in a prominent location on the user's device, along with additional input options. The user could then select from these phrases, or revert to other input methods. In actual use, one would also want such a system to incorporate speech recognition (ASR), but for the present we restrict ourselves to typed text — which is perfectly appropriate for some modes of interaction such as on-line social media domains. Using a corpus of 72 million words from American soap operas, we isolate features useful in predicting an appropriate set of responses for the previous utterance of an interlocutor. The main results of this work are a method that can automati-

cally produce appropriate responses to utterances in some cases, and an estimate of what percentage of dialog may be amenable to such techniques.

2 Previous Work

Alm et al. (1992) discuss how AAC technology can increase social interaction by having the utterance, rather than the letter or word, be the basic unit of communication. Findings from conversational analysis suggest a number of utterances common to conversation, including short conversational openers and closers (*hello*, *goodbye*), backchannel responses (*yeah?*), and quickfire phrases (*That's too bad.*). Indeed "small talk" is central to smooth-flowing conversation (King et al., 1995). Many modern AAC systems therefore provide canned small-talk phrases (Alm et al., 1993; Todman et al., 2008).

More complex conversational utterances are challenging to predict, and recent systems have used a variety of approaches to generate longer phrases from minimal user input. One approach relies on telegraphic input, where full sentences are constructed from a set of uninflected words, as in the Compansion system (McCoy et al., 1998). This system employs a semantic parser to capture the meaning of the input words and generates using the Functional Unification Formalism (FUF) system (Elhadad, 1991). One of the limitations of this approach is that information associated with each word is primarily hand-coded on the basis of intuition; as a result, the system cannot handle the problem of unrestricted vocabulary. Similar issues arise in semantic authoring systems (Netzer and Elhadad, 2006), where at each step of the sentence creation process, the system offers possible symbols for a small set of concepts, and the user can select which is intended.

Recent work has also tried to handle the complexity of conversation by providing full sentences with slots that can be filled in by the user. Dempster et al. (2010) define an ontology where pieces of hand-coded knowledge are stored and realized within several syntactic templates. Users can generate utterances by entering utterance types and topics, and these are filled into the templates. The Frametalker system (Higginbotham et al., 1999) uses contextual frames — basic sentences for different contexts — with a set vocabulary for each. The intuition behind this system is that there are typical linguistic structures for different situations and the kinds of words that the user will need to fill in will be semantically related to the context. Wisenburn and Higginbotham (2008) extend this technology using ASR on the speech of the interlocutor. The system extracts noun phrases from the speech and presents those noun phrases on the AAC device, with frame sentences that the user can then select. Thus, if the interlocutor says *Paris*, the AAC user will be able to select from phrases like *Tell me more about Paris* or *I want to talk about Paris*.

Other approaches provide a way for users to quickly find canned utterances. WordKeys (Langer and Hickey, 1998) allows users to access stored phrases by entering key words. This system approaches generation as a text retrieval task, using a lexicon derived from WordNet to expand user input to find possible utterances. Dye et al. (1998) introduce a system that utilizes scripts for specific situations. Although pre-stored scripts work reasonably well for specific contexts, the authors find (not unexpectedly) that a larger number of scripts are needed for the system to be generally effective.

3 The Soap Opera Corpus

In this work we attempt a different approach, developing a system that can learn appropriate responses to utterances given a corpus of conversations.

Part of the difficulty in automatically generating conversational utterances is that very large corpora of naturally occurring dialogs are non-existent. The closest such corpus is Switchboard (Godfrey and Holliman, 1997), which contains 2,400 two-sided conversations with about 1.4 million words. The interlocutors in Switchboard are not acquainted with each other and they are instructed to discuss a particular topic. While the dialogs are "natural" to a point, because they involve people who have never previously met, they are not particularly reflective of the kinds of conversations between intimates that we are interested in helping impaired users with.

We thus look instead to a corpus of scripted dialogs taken from American soap operas. The website `tvmegasite.net` contains soap opera scripts that have been transcribed by aficionados of the various series. The scripts include utterances marked

with information on which character is speaking, and a few dramatic cues. We downloaded 72 million words of text, with 5.5 million utterances. Soap opera series downloaded were: *All my Children*, *As the World Turns*, *The Bold and the Beautiful*, *Days of our Lives*, *General Hospital*, *Guiding Light*, *One Life to Live* and *The Young and the Restless*. The text was cleaned to remove HTML markup and other extraneous material, and the result was a set of 550,000 dialogs, with alternating utterances by (usually) two speakers. These dialogs were split 0.8/0.1/0.1 into training, development testing and testing portions, respectively. All results reported in this paper are on the *development test set*.

While soap operas may not be very representative of most people's lives, the corpus nonetheless has three advantages. First of all, the corpus is large. Second, the language tends to be fairly colloquial. Third, many of the dialogs take place between characters who are supposed to know each other well, often intimately; thus the topics might be more reflective of casual conversation between friends and intimates than the dialogs one finds in Switchboard.

4 Data Analysis, Feature Extraction and Utterance Prediction

Each dialog was processed using the Stanford Core NLP tools. The Stanford tools perform part of speech tagging (Toutanova et al., 2003), constituent and dependency parsing (Klein and Manning, 2003), named entity recognition (Finkel et al., 2005), and coreference resolution (Lee et al., 2011). From the output of the Stanford tools, the following features were extracted for each utterance: *word bigrams* (pairs of adjacent words); *dependency-head relations*, along with the type of dependency relation (basically, governors — e.g., verbs — and their dependents — e.g., nouns); *named entities* (persons, organizations, etc.); and *the whole utterance*. Extracted named entities include noun phrases that were explicitly tagged as named entities, as well as any phrases that were marked as coreferential with named entities. Thus if the pronoun *she* occurred in an utterance, and was marked as coreferential with a previous or following named entity *Amelia*, then the feature *Amelia* as a named entity was added for this utterance. We also include the whole utterance as a feature, which turns out to be the most useful predictor for an appropriate response to an input utterance.

The dialogs were divided into turns, with each turn consisting of one or more utterances. For our experiments, we are interested in predicting the *first utterance* of a turn (which in many cases may be the whole turn) *given features of all the utterances of the previous turns* — the exception being that for the whole sentence feature, only the last sentence of the previous turn is used. The method of using features of a turn to predict features of the next turn is related to the work reported in Purandare and Litman (2008), though their goal was to analyze dialog coherence rather than to predict the next utterance.

We are particularly interested in feature values that are highly skewed in their predictions, meaning that if the turn has a given value, then the first sentence of the next utterance is much more likely to have some values than others. A useful measure of this is the difference between the entropy of the predicted feature values f_i of a feature g:

$$H(g) = -\sum_{i=0}^{n} \log(p(f_i)) \cdot p(f_i) \qquad (1)$$

and the maximum possible entropy of g given n predicted features, namely:

$$H_{max}(g) = -\log(\frac{1}{n}) \qquad (2)$$

The larger the difference $H_{max}(g) - H(g)$, the more skewed the distribution.

For the purposes of this experiment and to keep the computation reasonably tractable, we computed the entropic values described above for like features: thus we used bigram features to predict bigram features, dependency features to predict dependency features, and so forth. We also filtered the output of the process so that each feature of the prior context had a minimum of 10 occurrences, and the entropy of the feature was no greater than 0.9 of the maximum entropy as defined above. For each feature value, the 2 most strongly associated values for the predicted utterance were stored.

To take a simple example (Figure 1) the bigram *'m fine* has a strong association with the bigrams *you 're* and *, I*, these co-occurring 486 and 464 times in the training corpus, respectively. For this feature, the

```
'm fine 8.196261 9.406976     you 're 486
'm fine 8.196261 9.406976     , i     464

you're kidding . __SENT 4.348040 4.852030
no. . __SENT 32
you're kidding . __SENT 4.348040 4.852030
i wish . __SENT 7
```

Figure 1: Examples of bigram and full-sentence features.

entropy is 8.20 and the maximum entropy is 9.41. Or consider a full-sentence feature *You're kidding*. This is strongly associated with the predicted sentence features *no.*. and *I wish.*.

Utterances in the training data were stored and associated with predicted features. In order to produce a rank-ordered list of possible responses to a test utterance, the features of the test utterance are extracted. For each of these features, the predicted features and their entropies are retrieved. Those training data utterances that match on one or more of these predicted features are retrieved in this step, and a score is assigned which is simply the sum of the predicted feature entropies. However, since we want to favor full-sentence matches, entropies for full-sentence matches are multiplied by a positive number (currently set to 100).

5 Experimental Results

5.1 Whole sentence prediction

The first question we were interested in is how often, based on the approach described here, one could predict a sentence that is close to what the speaker actually intended to say. For this purpose, we simply took as the gold standard the utterance that was written in the script for the speaker, and considered the prediction of the system described above, when it was able to make one. The prediction could be an exact match to what was actually said, something close enough to be a reasonable substitute, something appropriate given the context but not the one intended, or something that is wholly inappropriate.

In the ensuing discussion we will focus on whole sentence features, since these were the most useful for predicting reasonable whole sentences. We return to the use of other features in Section 5.2.

Some examples can be found in Figure 2. In each case, we give the final sentence of the previous turn, the actual utterance, and the two predicted utterances, along with the predicted utterances' scores and the counts with which they co-occurred in the training data with the previous utterance in question. For the first example *Really?*, the actual response was *Yeah*, and this was also the highest ranked response of the system. In the second example, the actual response was *I love you, too, baby doll*, whereas a response of the system was *I love you too*. While not exact, this is arguably close enough, and could be selected by an impaired user who did not wish to type the whole message. In the third example, the predictions *Yeah.* and *Ok.* do not substitute at all for the actual response.

```
PREV   really ?
ACTUAL yeah .
PRED   232.3099  yeah . __SENT 4
PRED   230.9528  mm-hmm . __SENT 3

PREV   love you .
ACTUAL i love you , too , baby doll .
PRED   83.4519 i love you , too . __SENT 3
PRED   74.1185 love you . __SENT 3

PREV   ok ?
ACTUAL i'm sorry , laurie , about j.r. ,
       about everything .
PRED   86.2623  yeah . __SENT 2
PRED   86.2623  ok . __SENT 2
```

Figure 2: Whole sentence prediction examples.

Of the 276,802 utterance-response pairs in the development test data, the system was able to make predictions for 9,794 cases, or 3.5%. Evaluating 9,794 responses is labor intensive, so two evaluations based on random samples were performed.

In the first, the authors evaluated a random sample of 455 utterance pairs, assigning the following scores to each response: **4** exact match; **3** equivalent meaning; **2** good answer but not the right one; **1** inappropriate. The results are given in Table 1, for the *best score* of the pair of responses generated. In other words, if the first response has a score of 2 and the second a score of 3, then the pair of responses will receive a score of 3: in that pair, there was one generated response that was close enough to use. From Table 1, we see that between 38% to 40.7% of the response pairs contained a response that was exact, or close enough to have the same meaning. 59.3% to 62% had at best a reasonable answer, but not the one intended. Finally, none contained only

Score	Judge 1		Judge 2	
Exact match	110	24.2%	109	24.0%
Equivalent meaning	63	13.8%	76	16.7%
Good answer (but wrong)	282	62.0%	270	59.3%
Inappropriate	0	0.0%	0	0.0%

Table 1: Judgments of a sample of 455 utterance pairs by the authors.

inappropriate answers: this is not surprising, given that all of the predicted responses were based on what was found in the training data, which one may assume involved largely felicitous interactions.

We also used Amazon's Mechanical Turk (AMT) to collect judgements from unbiased judges. Based on our previous evaluation, we expanded the *equivalent meaning* category into two more fine-grained categories, *essentially the same* and *similar meaning*, in order to capture phrases with slightly different connotations. This results in the 4-point scale in Table 2. Exact matches were found automatically before giving response pairs to Turkers, and account for a large portion of the data — 2,330 of the 9,794 response pairs, or 23.8%. For the remaining 76.2%, 138 participants were asked to judge how close the predicted response was to the actual response.

Each AMT participant was presented with six prompts (three entropy-based conversational turns and three chatbot-based conversational turns, discussed below). Each prompt listed the utterance, actual response, and predicted response. Two additional prompts with known answers were included to automatically flag participants who were not focusing on the task. Evaluation results are given in

4	**Essentially the same:**	They're pretty close, and mean basically the same thing.
3	**Similar meaning:**	They're similar, but the predicted response has a slightly different connotation from the actual response.
2	**Good answer, but not the right one:**	They're different, but the predicted response is still a reasonable response to the comment.
1	**Inappropriate:**	Different, and the predicted response is a totally unreasonable response to the comment.

Table 2: Four-point scale for AMT evaluation. Exact matches were found automatically.

Essentially the same	89	16.4%
Similar meaning	81	14.9%
Good answer (but wrong)	165	30.4%
Inappropriate	79	14.5%

Table 3: Evaluation results from AMT on a random sample of 414 predicted utterances (excluding exact matches).

Table 3. Percentages are multiplied by the proportion of results they represent (.762). Of the evaluated cases, we find that 31.3% of the predicted responses were judged to be essentially the same or similar to the actual response. 30.4% were judged to be a reasonable answer, and the remaining 14.5% were judged to be inappropriate.

Evaluation by AMT judges was thus much more favorable towards the prediction-based system than the authors' evaluation. Where the authors found 13.8%-16.7% to be essentially the same or similar, unbiased judges found just under a third of the data to meet these criteria. Coupled with the automatically detected exact matches, 55.1% of the predicted responses were found to be a reasonable approximation of (or exactly) the intended response. A smaller portion of the data was thought to be a good answer (but wrong), or wholly inappropriate.

5.2 Prediction with features plus a prefix of the intended utterance

It is of course not necessary for the system to predict the whole response without any input from the user. As with word prediction, the user might type a *prefix* of the intended utterance, and the system could then produce a small set of corresponding responses, among which would often be the one desired.

In order to evaluate such a scenario, we considered the shortest prefix of the actual intended response that would be consistent with a maximum of five sentences predicted from the features of the previous turn. Thus, we gathered the entire set of sentences from the training data that matched one or more of the predicted features, then began (virtually) typing the actual response. There are two possible outcomes. If the actual response is not in the set, then at some point the typed prefix will be consistent with none of the sentences in the set. In this worst case, the user would simply have to type the whole sentence (possibly using whatever word-completion

technology is already available on the device). But if the intended response is in the set, then at some point the set consistent with the prefix will be winnowed down to at most five members. The length of the prefix at that point, subtracted from the length of the intended sentence, is the keystroke savings.

Of the 276,802 utterances in the development test responses, 11,665 (4.2%) had a keystroke savings of greater than zero: thus, in 4.2% of cases, the intended utterance was to be found among the set of sentences consistent with the predicted features. The total keystroke savings was 102,323 characters out of a total of 8,725,508, or about 1%. While this is clearly small, note that it is over and above whatever keystroke savings one would gain by other methods, such as language modeling.

5.3 ALICE

A final experiment involved using a *chatbot* to generate responses. Previous approaches have used stored sentence templates that are generated based on keyword input from the user; a similar approach is used in a chatbot, where the input utterances are themselves triggers for the generated content. For this experiment, we used the publicly available ALICE (Wallace, 2012), which won the Loebner Prize (a Turing test) in 2000, 2001, and 2004. ALICE makes use of a large library of pattern-action pairs written in AIML (Artificial Intelligence Markup Language): if an input sentence matches a particular pattern, a response is generated by a rule that is associated with that pattern. ALICE follows conversational context by using a notion of TOPIC (what the conversation is currently about, based on keywords) and of THAT (the bot's previous utterance). Both are used along with the input utterance when selecting what next to say. In essence, ALICE is a much more sophisticated version of the 1960s Eliza program (Weizenbaum, 1966).

In order to use the chatbot for this task, we use an AIML interpreter (Stratton, 2010) on the most recent set of ALICE knowledge.[1] ALICE was given the utterances for each conversation in our development testing set, which allows the system to store some of the dialogue context under its THAT and TOPIC variables.

[1] http://code.google.com/p/aiml-en-us-foundation-alice/, retrieved February 2012.

Essentially the same	45	10.7%
Similar meaning	96	22.9%
Good answer (but wrong)	135	32.1%
Inappropriate	138	32.9%

Table 4: Evaluation results from AMT on a random sample of 414 chatbot utterances (excluding exact matches).

Example responses are given in Figure 3. As with the previous experiments, some responses are close to the actual intended message (first example in Figure 3). In some other cases (second example), the response is reasonable, though not the one intended. But in many cases, the response is too "cute", as in the examples on the righthand side.

Evaluation with AMT is given in Table 4, using the same scoring criteria as in Table 3. Exact matches are again automatically removed, and account for 142 responses (1.5%). For the remaining data, participants were asked to judge how close the chatbot response was to the actual response. Percentages are multiplied by the proportion of results they represent (.9855).

The chatbot is judged to produce less acceptable utterances than the entropy-based approach, with 10.7% essentially the same as the actual response, and 22.9% with similar meaning. Coupled with the automatically detected exact matches, 35% of the chatbot responses were found to be a reasonable approximation of (or exactly) the intended response. Over half of the data (65%) was thought to be a good answer (but wrong), or wholly inappropriate.

6 Discussion and Future Work

Of the two different methods for automatically generating responses, the entropy-based approach in particular produces responses ranging from similar to exact in a majority of the cases for which it can make a prediction. These are promising findings for expanding a system that populates possible responses from an input utterance.

An obvious limitation of the work presented here is that it is based on scripted data. More to the point, to build an effective dialog-based system for an impaired user, the best possible data to use would be data involving the user him/herself along with his or her interlocutors. Indeed, we are in the process

```
PREV    okay .
ACTUAL  okay .
ALICE   Great.

PREV    please don't leave .
ACTUAL  i'm right here .
ALICE   I'm not going anywhere.

PREV    you need to drink water .
ACTUAL  no , stay .
ALICE   I need to wait and see if the
        opportunity presents itself.

PREV    okay ?
ACTUAL  you were so mad at me
        lately , dixie .
ALICE   Dude!
```

Figure 3: Some responses generated by ALICE.

of collecting a corpus of such interactions from a small number of AAC users in the Portland, Oregon area. But the resulting corpora will obviously be tiny in comparison with the data used in the experiments here, in no small measure because of the extreme slowness with which most AAC users are able to communicate. What can be done about this? One thing would be to use the results of this work directly even if it does not model the particular user: even if it comes from soap opera dialogs, *Are you mad at me? No, I'm not mad at you*, still makes for a perfectly reasonable utterance/response pair. This, to some extent, counters potential objections that soap opera dialogs are not reflective of natural interactions. These kinds of pairs could be supplemented by whatever data we are able to learn from a particular user.

Even better, though, would be to collect large amounts of data from users *before* they become impaired. Many disorders, such as ALS, are often detected early, before they start to impair communication. In such cases, one could consider language-banking the user's interactions, and building a model of the ways in which the user interacts with other speakers, in order to get a good model of that *particular* user. While there are obviously privacy concerns, a person who knows that they will lose the ability to speak over time will likely be very motivated to try to preserve samples of their speech and language, assuming there exists technology that can use those samples to provide more sophisticated assistance when it becomes needed.

It may also be possible to use features from the text to generate utterances, similar to the telegraphic approaches to generation discussed in Section 2, but automatically learning words that can be used to generate appropriate responses to an utterance. As a first look at the feasibility of this approach, we use the Midge generator (Mitchell et al., 2012), rebuilding its models from the soap dialogues. Midge requires as input a set of nouns and then builds likely syntactic structure around them, and so we use the dialogues to predict possible nouns in response to an input utterance. For each <utterance, response> pair in the dialogues, we gather all utterance nouns n_u and all response nouns n_r. We then compute normalized pointwise mutual information (nPMI) for each n_u, n_r pair type in the corpus. Given a novel input utterance, we tag it to extract the nouns and create the set of highest nPMI nouns from the model. This is then input to Midge, which uses the set to generate present-tense declarative sentences. Some examples are given in Figure 4. We hope to expand on this approach in future work.

A further improvement is to take advantage of synonymy. There are many ways to convey the same basic message: *i am sick*, *i am not feeling well*, *i'm under the weather*, are all ways for a speaker to convey that he or she is not in the best of health. In the current system, these are all treated separately. Clearly what is needed is a way of recognizing that these are all *paraphrases* of each other. Fortunately, there has been a lot of progress in recent years on paraphrasing — see Ganitkevitch et al. (2011) for a recent example — and such work could in principle be adapted to the problem here. Indeed it seems likely that incorporating paraphrasing into the system will be a *major* source of improved coverage.

A limitation of the work described here is that it only models turn-to-turn interactions. Clearly discourse models need to have more memory than this, so features that relate to earlier turns would be needed. The downside is that this would quickly lead to data sparsity.

There are a variety of machine learning techniques that could also be tried, beyond the rather

Input: this is n't the same . this is not like anything i have been through before . i mean , how am i supposed to make it work with somebody who ...
Pred. nouns: strength, somebody
Output: strength comes with somebody

Input: i 've been a little bit too busy to socialize . i did have an interesting conversation with your sister , however .
Pred. nouns: bit, conversation, sister
Output: a bit about this conversation with sister

Figure 4: Generating with nPMI: Creating syntactic structure around likely nouns.

simple methods employed in this work. For example, particular classes of response types, comprising a variety of related utterances, may be predictable using the extracted features.

Finally, we have assumed for this discussion that the AAC system is only within the control of the impaired user. There is no reason to make that assumption in general: many AAC situations in real life involve a helper who will often *co-construct* with the impaired user. Such helpers usually know the impaired user very well and can often make reasonable guesses as to the whole utterance intended by the impaired user. Recent work reported in Roark et al. (2011) suggests one way in which the results of a language modeling system and those of a human co-constructor may be integrated into a single system, and such an approach could easily be applied here.

7 Conclusions

We have proposed and evaluated an approach to whole utterance prediction for AAC. While the approach is fairly simple, it is able to generate correct or at least reasonable responses in some cases. Such a system could be used in conjunction with other techniques, such as language-model-based prediction, or co-construction. One of the potentially useful side-effects of this work is an estimate of what percentage of interactions in a dialog are likely to be easily handled by such techniques. In other words, how many interactions in dialog are sufficiently predictable that a system could have a reasonable guess as to what a speaker is going to say given the previous context? A rough estimate based on what we have found here is something on the order of 3.5%-4.0%. Obviously this does not mean that the system will always make the right prediction: a reasonable response to *congratulations on your promotion* would often be *thank you*, but a speaker may wish to say something else. But what it does mean is that in about 3.5%-4.0% of cases, one has a reasonable chance of being able to guess. This percentage is certainly small, and one might be inclined to conclude that the approach does not work. On the other hand, it is important to bear in mind that not all percentages are created equal. Rapid responses to basic phrases (e.g. *Are you mad at me?* $\rightarrow$ *No, I'm not mad at you*), could help with the perceived flow of conversation, even if they do not occur that frequently.

As we noted at the outset, whole utterance prediction is an area that has received increased interest in recent years, because of its potential to speed communication, and its contribution to increasing the naturalness of conversational interactions. When coupled with gains in utterance generation achieved by other methods, automatically generating utterances can further the range of comments and responses available to AAC users. The work reported here is a small contribution towards this goal.

Acknowledgments

This work was supported under grant NIH-K25DC011308. Sproat thanks his K25 mentor, Melanie Fried-Oken, for discussion and support. We also thank four anonymous reviewers, as well as the audience at a Center for Spoken Language Understanding seminar, for their comments.

References

N. Alm, J. L. Arnott, and A. F. Newell. 1992. Prediction and conversational momentum in an augmentative communication system. *Communications of the ACM*, 35(5):46–57.

N. Alm, J. Todman, Leona Elder, and A. F. Newell. 1993. Computer aided conversation for severely physically impaired non-speaking people. *Proceedings of INTERCHI '93*, pages 236–241.

Bruce Baker. 1990. Semantic compaction: a basic technology for artificial intelligence in AAC. In *5th Annual Minspeak Conference*.

J. J. Darragh, I. H. Witten, and M. L. James. 1990. The reactive keyboard: A predictive typing aid. *Computer*, 23(11):41–49.

Martin Dempster, Norman Alm, and Ehud Reiter. 2010. Automatic generation of conversational utterances and narrative for augmentative and alternative communication: A prototype system. *Proceedings of the Workshop on Speech and Language Processing for Assistive Technologies (SLPAT)*, pages 10–18.

R. Dye, N. Alm, J. L. Arnott, G. Harper, and A Morrison. 1998. A script-based AAC system for transactional interaction. *Natural Language Engineering*, 4(1):57–71.

Michael Elhadad. 1991. FUF: The universal unifer-user manual version 5.0. Technical report.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 363–370.

Juri Ganitkevitch, Chris Callison-Burch, Courtney Napoles, and Benjamin van Durme. 2011. Learning sentential paraphrases from bilingual parallel corpora for text-to-text generation. *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

John Godfrey and Edward Holliman. 1997. Switchboard-1 release 2. Linguistic Data Consortium, Philadelphia.

D. J. Higginbotham, D. P. Wilkins, G. W. Lesher, and B. J. Moulton. 1999. Frametalker: A communication frame and utterance-based augmentative communication device. Technical Report.

D. Jeffery Higginbotham. 1992. Evaluation of keystroke savings across five assistive communication technologies. *Augmentative and Alternative Communication*, 8:258–272.

Julia King, Tracie Spoeneman, Sheela Stuart, and David Beukelman. 1995. Small talk in adult conversations: Implications for AAC vocabulary selection. *Augmentative and Alternative Communication*, 11:260–264.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. *Proceedings of the 41st Meeting of the Association for Computational Linguistics (ACL)*, pages 423–430.

S. Langer and M. Hickey. 1998. Using semantic lexicons for full text message retrieval in a communication aid. *Natural Language Engineering*, 4(1):41–55.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. *Proceedings of the CoNLL-2011 Shared Task*.

J. Li and G. Hirst. 2005. Semantic knowledge in word completion. In *Proceedings of the 7th International ACM Conference on Computers and Accessibility*.

K. McCoy, C. A. Pennington, and A. L. Badman. 1998. Compansion: From research prototype to practical integration. *Natural Language Engineering*, 4(1):73–95.

Kathleen F. McCoy, Jan L. Bedrosian, Linda A. Hoag, and Dallas E. Johnson. 2007. Brevity and speed of message delivery trade-offs in augmentative and alternative communication. *Augmentative and Alternative Communication*, 23(1):76–88.

Margaret Mitchell, Jesse Dodge, Amit Goyal, Kota Yamaguchi, Karl Sratos, Xufeng Han, Alysssa Mensch, Alex Berg, Tamara L. Berg, and Hal Daumé III. 2012. Midge: Generating image descriptions from computer vision detections. *Proceedings of EACL 2012*.

Y. Netzer and M. Elhadad. 2006. Using semantic authoring for Blissymbols communication boards. *Proceedings of HLT 2006*, pages 105–108.

Amruta Purandare and Diane Litman. 2008. Analyzing dialog coherence using transition patterns in lexical and semantic features. In *FLAIRS Conference*, pages 195–200.

Brian Roark, Andrew Fowler, Richard Sproat, Christopher Gibbons, and Melanie Fried-Oken. 2011. Towards technology-assisted co-construction with communication partners. *Proceedings of the Workshop on Speech and Language Processing for Assistive Technologies (SLPAT)*.

Cort Stratton. 2010. PyAIML, a Python AIML interpreter. http://pyaiml.sourceforge.net/.

J. Todman, A. Norman, J. Higginbotham, and P. File. 2008. Whole utterance approaches in AAC. *Augmentative and Alternative Communication*, 24(3):235–254.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. *Proceedings of HLT-NAACL*, pages 252–259.

K. Trnka, D. Yarrington, K.F. McCoy, and C. Pennington. 2006. Topic modeling in fringe word prediction for AAC. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 276–278.

K. Trnka, D. Yarrington, J. McCaw, K.F. McCoy, and C. Pennington. 2007. The effects of word prediction on communication rate for AAC. In *Proceedings of HLT-NAACL; Companion Volume, Short Papers*, pages 173–176.

H. Trost, J. Matiasek, and M. Baroni. 2005. The language component of the FASTY text prediction system. *Applied Artificial Intelligence*, 19(8):743–781.

Richard Wallace. 2012. A.L.I.C.E. (Artificial Linguistic Internet Computer Entity). http://www.alicebot.org/.

T. Wandmacher and J.Y. Antoine. 2007. Methods to integrate a language model with semantic information for a word prediction component. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 506–513.

Joseph Weizenbaum. 1966. Eliza – a computer program for the study of natural language communication between man and machine. *Proceedings of the ACM*, 9(1).

Bruce Wisenburn and D. Jeffery Higginbotham. 2008. An AAC application using speaking partner speech recognition to automatically produce contextually relevant utterances: Objective results. *Augmentative and Alternative Communication*, 24(2):100–109.