Topic Modeling in Fringe Word Prediction for AAC

Keith Trnka, Debra Yarrington,
Kathleen McCoy
Computer Science Department
University of Delaware
Newark, DE 19716

{trnka,yarringt,mccoy}@cis.udel.edu

Christopher Pennington
AgoraNet, Inc.
314 E. Main St., Suite 1
Newark, DE 19711

penningt@agora-net.com

ABSTRACT

Word prediction can be used for enhancing the communication ability of persons with speech and language impairments. In this work, we explore two methods of adapting a language model to the topic of conversation, and apply these methods to the prediction of fringe words.

Categories and Subject Descriptors:
I.7.m Document and Text Processing: Miscellaneous

General Terms: Algorithms

Keywords: Word prediction, topic modeling, language modeling, AAC

1. INTRODUCTION

Augmentative and Alternative Communication (AAC) is the field of research concerned with finding ways to help those with speech difficulties communicate more easily and completely. Today there are approximately 2 million people in the United States with some form of communication difficulty. One means to help ease communication is the use of an electronic communication device, which may have synthetic speech as output. However, one issue in using an AAC device is communication rate. Whereas speaking rate is estimated at 180 words per minute (wpm), many AAC users' communication rates are lower than 15 wpm [3, 7, 16]. Thus one goal of developers is to find ways to increase the rate of communication, by making AAC devices easier to use and more intelligent.

Some researchers have attempted to speed communication rate by providing quick access to the *core vocabulary* – the relatively small set of frequently used words. Methods for doing this include abbreviation expansion and iconic methods such as semantic compaction [1]. In contrast, in this work we attempt to speed access to the much larger set of words often called *fringe vocabulary*. This set is of interest because although each individual word occurs less frequently, the set of fringe words on the whole is very significant.

Suppose that the user wants to enter "I want a home in the country." After typing, "I want a h", they might see something like shown below. The system has created a *prediction window* containing the five words that it thinks the user may be trying to type. In this example, the user can press F5 to complete the word "home" and the system will

enter the word with a space afterwards. So in this example, the user needed 2 keystrokes to enter what would normally take 5 keystrokes.

I want a	h	
	hundred	(F1)
	half	(F2)
	house	(F3)
	hard	(F4)
	home	(F5)

It is difficult to judge how much word prediction can speed communication rate. Much of this determination is dependent on the accuracy of the prediction method, the characteristics of the user, such as their physical and cognitive abilities, and the characteristics of the user interface, such as where the prediction list is displayed and how a word in the list is selected. Here, the prediction method is evaluated separately from the rest of a word prediction system by simulating what a user would type in a conversation if he/she were taking full advantage of the prediction list. This theoretical evaluation measures the percentage of keystrokes that were saved by word prediction over typing out every character.

In this paper we first describe related work and give some background in statistical approaches to word prediction. We present approaches to topic modeling and compare the results of topic modeling to a baseline method. For a more thorough account of this work, visit
http://www.cis.udel.edu/fringe/.

2. RELATED WORK

Several previous researchers have used n-gram models in word prediction for AAC [4, 5, 12, 18]. For example, Lesher et al. [12] show the impact of increasing training set size and going from unigrams to bigrams (47% to 54.7%) to trigrams (another .8%). These evaluations used a window size of 6.

Other researchers have integrated grammatical information into n-gram word prediction systems. Garay-Vitoria and Gonzalez-Abascal [10] integrated a statistical chart parser, while Fazly and Hirst [8] and Copestake [7] used part-of-speech (POS) tagging. These yielded improvements of 1-5% keystroke savings.

There have been several attempts at topic modeling in the language modeling community, particularly for speech recognition [2, 14, 17, 6, 9, 13]. Some of the evaluations of topic modeling have found different variants of it to be very beneficial [2, 14, 9]. Lesher and Rinkus [13] is an attempt at topic modeling for word prediction, but does not use dynamic topic modeling like [9, 2] and this work.

Window	Bigrams	Trigrams	Method A	Method B
1	41.5%	42.3%	43.1%	42.5%
2	50.6%	51.1%	52.3%	51.4%
3	54.7%	55.1%	56.4%	55.4%
4	57.0%	57.3%	58.7%	57.7%
5	58.6%	58.8%	60.2%	59.1%
6	59.8%	60.0%	61.4%	60.3%
7	60.6%	60.8%	62.2%	61.1%
8	61.3%	61.5%	62.9%	61.8%
9	61.9%	62.0%	63.5%	62.3%
10	62.4%	62.5%	64.0%	62.8%

Table 1: The keystroke savings of topic modeling is shown compared to a bigram and trigram baseline.

3. METHODS

Like several of the aforementioned word prediction researchers, we use n-gram methods for language modeling. Our baseline word prediction methods use bigram and trigram-based n-gram models with backoff with Good-Turing smoothing, the current best practice in statistical language modeling according to Manning and Schütze [15]. Additionally, we incorporate a special unigram model for the first word of each sentence. In word prediction, these language models are used to rank all the words that the user could possibly be typing. The top W words are presented to the user, where W is the prediction window size.

Statistical approaches require a collection of text to construct a language model. Ideally, our corpus would be a large collection of conversations involving one or more people using an AAC system. Such a corpus is unavailable, so we follow [13] in using the Switchboard corpus, which is a collection of telephone conversations and their transcriptions.[1] The training section contains a randomly pre-selected 2217 conversations and the testing section contains the remaining 221 conversations. We perform preprocessing to remove some speech repairs in accordance with Hindle [11]. These editing rules bring the Switchboard conversations closer to what we envision an AAC user would type.

3.1 Evaluation

We compare the number of keystrokes required for a user taking full advantage of our word prediction system to the number of keystrokes required to enter each character of the conversation. We use *immediate prediction* for our evaluations, which allows use of the prediction list before the first character of a word has been entered. We assume that one keystroke is required to "speak" each turn of input and that a space is automatically inserted after a word is selected from the prediction list.

$$KS = \frac{keys_{normal} - keys_{withprediction}}{keys_{normal}} * 100\%$$

Because we are interested in the prediction of fringe words, our evaluations are measured on fringe words only. Core words are excluded from the list of predictions. The particular core vocabulary we chose is available from the AAC Centers at the University of Nebraska at Lincoln, available from http://aac.unl.edu/. We used the "Young Adult Conversation Regular" core vocabulary list, as it is the most similar to the type of conversations in the Switchboard corpus.

---

[1]The Switchboard transcriptions were available from http://www.isip.msstate.edu/projects/switchboard/

4. TOPIC MODELING

The goal of topic modeling is to identify the current topic of conversation, then increase the probability of related words and decrease the probability of unrelated words. Some words will be unaffected by topic modeling, such as function words, which are used similarly in all topics. It is for this reason that we chose to improve fringe word prediction with topic modeling: we feel that topic modeling specifically improves fringe word prediction.

Researchers are consistent in representing a topic by creating a collection of representative text of the topic. However, researchers differ on the best way to organize a collection of topics. Some researchers have created a hierarchical collection of topics [9], while others have created a disjoint set of topics [14, 2, 17]. We feel that the primary lure of a hierarchical approach, the ability to generalize, can be captured in the set approach as well, by giving varying weight to all topics and not just the most likely topic. For this reason, we represent topics as disjoint sets of conversations.

The current topic of conversation must be identified from the part of the conversation that has taken place so far, and updated periodically in the conversation. Thus, we must devise a representation for a partial conversation for assessing the similarity of the conversation to each topic. In representing the conversation so far, we choose to implement an exponentially decayed cache, like [2], using TF-IDF values rather than raw frequencies. This follows the work of Mahajan et. al. [14] in considering the inverse document frequency of a word as proportional to its utility in identifying the current topic. Because our approach is for topic identification, we ignore words that occur in 85% or more of the topics, with the intuition that such words are irrelevant to selection of topic. As a step to convert our model of the current conversation to a model of the current topic, we compute the document similarity between the cache and the unigram model for each topic. We chose to use the cosine metric, following [9].

Given that we have computed a similarity score between each topic and the current conversation, there are two main variations on how to construct a new language model. Mahajan et. al. [14] implemented a k-nearest solution, constructing the topic model from the most similar k topics. Each topic's language model was weighted equally for their experiments. Instead, we chose to follow Florian and Yarowsky's approach [9]. They expand the probability for a word (w) given a history (h) as follows:

$$P(w \mid h) = \sum_{t \in topics} P(t \mid h) * P(w \mid t, h)$$

$P(w \mid t, h)$ is simply the probability of w taken from the language model constructed for topic t. The probability of the topic is estimated as follows:

$$P(t \mid h) \approx \frac{S(t, h)}{\sum_{t' \in topics} S(t', h)}$$

where $S(t, h)$ is the cosine similarity of the topic to the current part of the conversation.

4.1 Method A

Our first method of topic modeling is most similar in spirit to the work of Mahajan et. al. [14] and Florian and Yarowsky [9]. In training, a bigram model is computed for

each topic in Switchboard. In testing, the cache representation of the current conversation is compared against the unigram representation of each topic and similarity scores are computed. The similarity scores are then used to weight the frequencies obtained from each topic in a linear interpolation. Then this interpolated bigram model is used to compute the probabilities used for word prediction.

Topic modeling shows a sizable improvement over the the bigram baseline: 1.6% – 1.7%. We've included the comparison to a bigram baseline because it is the most natural baseline in terms of language understanding. However, a trigram baseline is also a natural comparison when considering that it can run with the same or less computational resources than topic modeling. When compared against the trigram baseline, the topic model gives 0.8% – 1.5% improvement.

4.2 Method B

Our second method of topic modeling is more similar to the work of Bellegarda [2]. Like Bellegarda, we compute topic-dependent unigram probabilities. These topic-dependent probabilities are multiplied with probabilities from a trigram backoff model. Additionally, we weight the topic component with a tuning parameter. After manual tuning on two conversations, we found that $\alpha = .15$ worked well.

Method B is an improvement over a trigram baseline, but only a minor improvement. We feel that the problem is that a low α value was necessary to avoid overriding the word preference that is due to context, but that it also reduced the ability of the overall model to adapt to a particular topic.

4.3 Comparison

Method A offers an additional 1% or more keystroke savings over Method B for most window sizes. This is due to the low weight of the tuning parameter for Method B. However, as previously mentioned, the low weight was necessary. Additionally, notice that Method A becomes comparatively better as the window size is increased. The trigram model component in Method B can be thought of as a stronger source of knowledge than the interpolated bigram model of Method A. Because of this, when the trigram history exists in the language model, Method B's predictions are more accurate. However, because the trigram model is sparse, it can only contribute to the top few predictions. Thus, it has a much greater effect on the top few window sizes.

For real world systems, however, absolute performance is not the only factor. The computational demands of each approach are often considered when selecting a practical solution. The trigram baseline processed at 1,325 words per minute (wpm). Method A processed conversations in testing at 32 wpm and Method B processed 1,267 words per minute. Method B uses barely more processing time than the trigram baseline model.

5. CONCLUSIONS

Topic modeling can be implemented in many different ways. We've demonstrated two such methods for topic modeling: one for computationally limited devices and another for computationally rich devices. Both methods show a clear improvement over a trigram model with backoff. Before the advent of word prediction, a user would've pressed 6.4 keys per fringe word on average. Now, with topic modeling for word prediction, only 2.5 keys per word are required.

7. REFERENCES

[1] B. Baker. Minspeak. *Byte*, pages 186–202, 1982.

[2] J. Bellegarda. Large vocabulary speech recognition with multispan language models. *IEEE Trans. On Speech and Audio Processing*, 8(1), 2000.

[3] D. R. Beukelman and P. Mirenda. *Augmentative and alternative communication: Management of severe communication disorders in children and adults*. P.H. Brookes Pub. Co., 1998.

[4] L. Boggess. Two simple prediction algorithms to facilitate text production. In *Proceedings of the second conference on Applied natural language processing*, pages 33–40, Morristown, NJ, USA, 1988. Association for Computational Linguistics.

[5] A. Carlberger, J. Carlberger, T. Magnuson, M. S. Hunnicutt, S. Palazuelos-Cagigas, and S. A. Navarro. Profet, a new generation of word prediction: An evaluation study. In *Proceedings of Natural Language Processing for Communication Aids*, 1997.

[6] S. Chen, K. Seymore, and R. Rosenfeld. Topic adaptation for language modeling using unnormalized exponential models. In *Proc. Int'l Conf. on Acoustics, Speech and Signal Processing*, 1998.

[7] A. Copestake. Augmented and alternative nlp techniques for augmentative and alternative communication. In *Proceedings of the ACL workshop on Natural Language Processing for Communication Aids*, 1997.

[8] A. Fazly and G. Hirst. Testing the efficacy of part-of-speech information in word completion. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, 2003.

[9] R. Florian and D. Yarowsky. Dynamic nonlocal language modeling via hierarchical topic-based adaptation. In *Proceedings of ACL'99*, pages 167–174, 1999.

[10] N. Garay-Vitoria and J. González-Abascal. Intelligent word-prediction to enhance text input rate. In *Proceedings of the second international conference on Intelligent User Interfaces*, 1997.

[11] D. Hindle. Deterministic parsing of syntactic non-fluencies. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, 1983.

[12] G. Lesher, B. Moulton, and J. Higgonbotham. Effects of ngram order and training text size on word prediction. In *Proceedings of the RESNA '99 Annual Conference*, 1999.

[13] G. Lesher and G. Rinkus. Domain-specific word prediction for augmentative communication. In *Proceedings of the RESNA '01 Annual Conference*, 2001.

[14] M. Mahajan, D. Beeferman, and X. D. Huang. Improved topic-dependent language modeling using information retrieval techniques. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1999.

[15] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 2000.

[16] A. Newell, S. Langer, and M. Hickey. The rôle of natural language processing in alternative and augmentative communication. *Natural Language Engineering*, 4(1):1–16, 1996.

[17] K. Seymore and R. Rosenfeld. Using story topics for language model adaptation. In *Proceedings of Eurospeech '97*, pages 1987–1990, Rhodes, Greece, 1997.

[18] A. L. Swiffin, J. A. Pickering, J. L. Arnott, and A. F. Newell. Pal: An effort efficient portable communication aid and keyboard emulator. In *Proceedings of the 8th Annual Conference on Rehabilitation Techonology*, pages 197–199, 1985.