Tom Stocky, Alexander Faaborg, Henry Lieberman. (2004) "A Commonsense Approach to Predictive Text Entry." *Proceedings of Conference on Human Factors in Computing Systems.* April 24-29, Vienna, Austria.

A Commonsense Approach to Predictive Text Entry

Tom Stocky, Alexander Faaborg, Henry Lieberman
MIT Media Laboratory
20 Ames St., Bldg E15
Cambridge, MA 02139 USA
{tstocky, faaborg, lieber}@media.mit.edu

ABSTRACT

People cannot type as fast as they think, especially when faced with the constraints of mobile devices. There have been numerous approaches to solving this problem, including research in augmented input devices and predictive typing aids. We propose an alternative approach to predictive text entry based on commonsense reasoning. Using OMCSNet, a large-scale semantic network that aggregates and normalizes the contributions made to Open Mind Common Sense (OMCS), our system is able to show significant success in predicting words based on their first few letters. We evaluate this commonsense approach against traditional statistical methods, demonstrating comparable performance, and suggest that combining commonsense and statistical approaches could achieve superior performance. Mobile device implementations of the commonsense predictive typing aid demonstrate that such a system could be applied to just about any computing environment.

Author Keywords

Predictive Interfaces, Open Mind Common Sense, Typing Aids.

ACM Classification Keywords

H.5.2 User Interfaces: *input strategies*.
I.2.7 Natural Language Processing: *text analysis*.

INTRODUCTION

People cannot type as fast as they think. As a result, they have been forced to cope with the frustration of slow communication, particularly in mobile devices. In the case of text entry on mobile phones, for example, users typically have only twelve input keys, so that to simply write “hello” requires thirteen key taps.

Predictive typing aids have shown some success, particularly when combined with algorithms that can disambiguate words based on single-tap entry. Past approaches to predictive text entry have applied text compression methods (e.g., [9]), taking advantage of the high level of repetition in language.

Similar approaches have applied various other statistical models, such as low-order word *n*-grams, where the probability of a word appearing is based on the *n*-1 words preceding it. Inherently, the success of such models depends on their training set corpora, but the focus has largely been on the statistics rather than the knowledgebase on which they rely.

We have chosen to focus on the knowledgebase issue, and propose an alternative approach based on commonsense reasoning. This approach performs on par with statistical methods and is able to anticipate words that could not be predicted using statistics alone. We introduce this commonsense approach to predictive text entry not as a substitute to statistical methods, but as a complement. As words predicted by the commonsense system tend to differ from those predicted by statistical methods, combining these approaches could achieve superior results to the individual performance of either.

RELATED WORK

Efforts to increase the speed of text entry fall into two primary categories: (1) new means of input, which increase efficiency by lessening the physical constraints of entering text, and (2) predictive typing aids, which decrease the amount of typing necessary by predicting completed words from a few typed letters.

Means of Input

Augmented keyboards have shown improvements in efficiency, both physical keyboards [1] and virtual [11]. In cases where the keyboard is constrained to a less efficient layout, disambiguation algorithms have demonstrated success in increasing efficiency [7].

Others have looked at alternate modalities, such as speech and pen gesture. Such modalities are limited by similar physical constraints to keyboard entry. And while speech recognition technology continues to improve, it is currently less efficient and less “natural” than keyboard entry [4].

Reducing the physical constraints around entering text is

CHI 2004, April 24–29, 2004, Vienna, Austria.
ACM 1-58113-703-6/04/0004.

extremely valuable, and we view predictive typing aids as a means to solving another part of the problem.

Predictive Typing Aids

One of the first predictive typing aids was the Reactive Keyboard [2], which made use of text compression methods [9] to suggest completions. This approach was statistically driven, as have been virtually all of the predictive models developed since then. Statistical methods generally suggest words based on:

1. Frequency, either in the context of relevant corpora or what the user has typed in the past; or
2. Recency, where suggested words are those the user has most recently typed.

Such approaches reduce keystrokes and increase efficiency, but they make mistakes. Even with the best possible language models, these methods are limited by their ability to represent language statistically. In contrast, by using commonsense knowledge to generate words that are semantically related to what is being typed, text can be accurately predicted where statistical methods fail.

PREDICTING TEXT USING COMMON SENSE

Commonsense reasoning has previously demonstrated its ability to accurately classify conversation topic [3]. Using similar methods, we have designed a predictive typing aid that suggests word completions that make sense in the context of what the user is writing.

Open Mind Common Sense

Our system's source of commonsense knowledge is OMCSNet [6], a large-scale semantic network that aggregates and normalizes the contributions made to Open Mind Common Sense (OMCS) [8]. OMCS contains a wide variety of knowledge of the form "tennis is a sport" and "to play tennis you need a tennis racquet." OMCSNet uses this knowledge – nearly 700,000 English sentences contributed by more than 14,000 people from across the web – to create more than 280,000 commonsensical semantic relationships.

It would be reasonable to substitute an *n*-gram model or some other statistical method to convert OMCS into relationships among words; the key is starting from a corpus focused on commonsense knowledge.

Using OMCSNet to Complete Words

As the user types, the system queries OMCSNet for the semantic context of each completed word, disregarding common stop words. OMCSNet returns the context as a list of phrases, each phrase containing one or more words, listing first those concepts more closely related to the queried word. As the system proceeds down the list, each word is assigned a score:

$$\text{score} = \frac{1}{\log_5(5+n)}$$

The variable *n* increments as the system works through the phrases in the context, so that the word itself (*n*=0) receives a score of 1.0, the words in the first phrase (*n*=1) receive a score of 0.90, those in the second phrase 0.83, and so on. Base 5 was selected for the logarithm as it produced the best results through trial-and-error. A higher base gives too much emphasis to less relevant phrases, while a lower base undervalues too many related phrases.

The scored words are added to a hash table of potential word beginnings (various letter combinations) and completed words, along with the words' associated total scores. The total score for a word is equal to the sum of that word's individual scores over all appearances in semantic contexts for past queries. As the user begins to type a word, the suggested completion is the word in the hash table with the highest total score that starts with the typed letters.

In this way, words that appear multiple times in past words' semantic contexts will have higher total scores. As the user shifts topics, the highest scored words progressively get replaced by the most common words in subsequent contexts.

EVALUATION

We evaluated this approach against the traditional frequency and recency statistical methods. Our evaluation had four conditions:

1. Language Frequency, which always suggested the 5,000 most common words in the English language (as determined by [10]);
2. User Frequency, which suggested the words most frequently typed by the user;
3. Recency, which suggested the words most recently typed by the user; and
4. Commonsense, which employed the method described in the previous section.

These conditions were evaluated first over a corpus of emails sent by a single user, and then over topic-specific corpora.

Each condition's predicted words were compared with those that actually appeared. Each predicted word was based on the first three letters typed of a new word. A word was considered correctly predicted if the condition's first suggested word was exactly equal to the completed word. Only words four or more letters long were considered, since the predictions were based on the first three letters.

Email Corpus

As predictive text entry is especially useful in mobile devices, we compiled an initial corpus that best approximated typical text messaging on mobile devices. This initial corpus consisted of a single user's sent emails over the past year. We used emails from only one user so

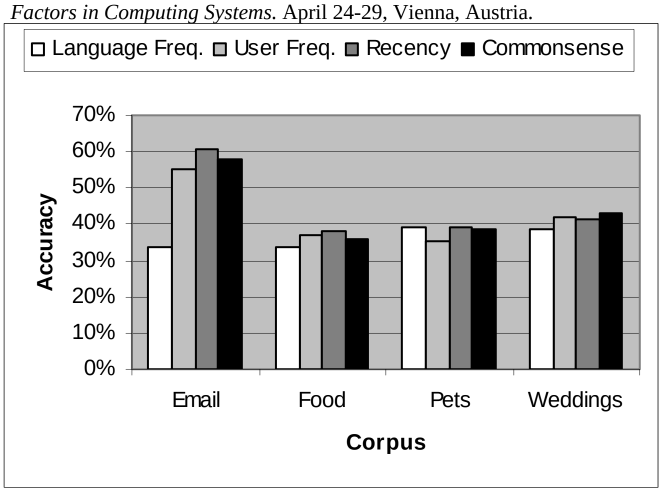


Figure 1. Accuracy of four conditions across various corpora.

that the corpus would be more suitable for the User Frequency and Recency conditions. There were 5,500 emails in total, consisting of 1.1M words, 0.6M of which were four or more letters long.

The results showed that Recency performed best, with an overall accuracy of 60.9%, followed by Commonsense at 57.7%, User Frequency at 55.1% and Language Frequency at 33.4%.

Overall, the performance of the commonsense approach was on par with the other conditions. Upon further analysis, it became clear that our system performed better relative to the other conditions when there was better coverage of the current topic in OMCS. Many of the emails were rather technical in nature, on topics scarcely mentioned in the commonsense database. By OMCS' very nature, its broad knowledgebase is not evenly distributed over all topics, so some topics experience more in-depth coverage than others.

With this in mind, we evaluated the four conditions on three additional corpora, which represented areas where OMCS had fairly significant coverage.

Topic-Specific Corpora

Evaluation was run over three additional corpora representing topics covered fairly well by OMCS:

1. *Food*: 20 articles from Cooking.com, selected at random – 10,500 total words, of which 6,500 were four or more letters long.
2. *Pets*: 20 articles from PetLifeWeb.com, selected at random – 10,500 total words, of which 6,000 were four or more letters long.
3. *Weddings*: 20 articles from WeddingChannel.com, selected at random – 16,500 total words, of which 10,000 were four or more letters long.

The results (summarized in Figure 1) showed once again that the commonsense approach was on par with the other conditions, performing best on the weddings corpus, where, of the three corpora, OMCS has the best coverage.

Where the Commonsense Approach Excels

Once again, we completed a detailed analysis of where the commonsense approach performed best and worst relative to the other conditions. Our system performed best (as much as 11.5% better on a 200 word section than the next best method) in cases of low word repetition, especially at times when the words selected were somewhat uncommon, as judged by the words' ranking in [10].

The following excerpt from the data illustrates this point:

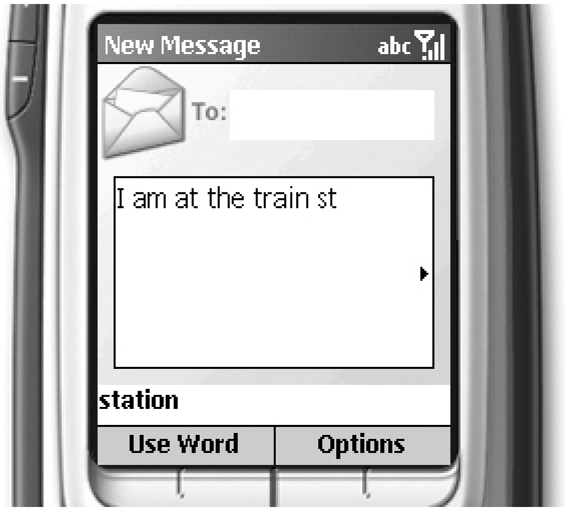
> "I spoke to my roommate -- sorry the rent isn't on time, he said he did pay it right at the end of last month"

In this case, there are several words that the commonsense system is able to predict correctly, while the others are not. Based on two of the first words typed – "spoke" and "roommate" – the system predicts three of the words that follow – "rent," "time," and "right." Those words, in turn, allow the prediction of "last" and "month." In total, of the last eight words four or more letters long, the commonsense system correctly predicts six (75%) of them, based only on two typed words and the predicted words themselves.

IMPLEMENTATION

The commonsense predictive text entry system was originally implemented on the Java 2 Platform, Standard Edition (J2SE), making use of the OMCSNet Java API.

Similar versions were implemented on a Motorola MPx200 Smartphone and a Pocket PC, using C# and the .NET Compact Framework, as well as on a Nokia 6600, using the Java 2 Platform, Micro Edition (J2ME) supporting MIDP (Mobile Information Device Profile) 1.0. Due to memory constraints, these versions used a subset of OMCSNet – approximately 10,000 nodes for the mobile phone implementations and 20,000 nodes for the Pocket PC



I am at the train st

station

Figure 2. Screenshot of Smartphone implementation.

version. Next generation devices will not have such memory constraints, and current constraints can be overcome with the use of external memory cards.

The system serves as a predictive typing aid that predicts word completions. Once the user has typed a two-letter word beginning, the system suggests the most relevant completed word. The user can then accept that suggestion, or can continue typing, which may result in a new predicted word completion based on the new letters.

These mobile device implementations demonstrate the feasibility of applying a commonsense system to just about any computing environment.

DISCUSSION AND FUTURE WORK

It is clear that commonsense knowledge is useful for predictive typing aids. While the system's performance is on par with statistical methods, what is more important is that the words predicted using common sense differ significantly from the other conditions. This suggests that the question is therefore not which method to use but how to *combine* the methods effectively, to exceed the performance of any individual method.

Combining Commonsense and Statistical Methods

One technique for combining commonsense and statistical methods would be to treat the contributions of each individual approach as multiple hypotheses. These hypotheses could then be weighted based on user behavior, as the system learns which methods are performing better in different contexts. The metric for tracking user behavior could be as simple as monitoring the number of accepted or rejected suggestions. This approach has the added benefit of gathering data about when different approaches work best, valuable information as predictive text entry reaches higher performance thresholds.

Phrase Completion

The current focus of our commonsense system is word completion. This does not take full advantage of the semantic links that OMCSNet can provide among concepts. As demonstrated by [5], commonsense knowledge is unique in its ability to understand context in language and semantic relationships among words. Commonsense knowledge is well suited for phrase expansion, which would allow a predictive text entry system based on commonsense to effectively predict phrase completions.

Natural Language Processing

This first evaluation was meant to serve as a baseline comparison. As such, none of the conditions made use of language models or part of speech taggers. Clearly, these would have improved performance across all conditions. In designing future predictive typing aids, it would be worth exploring how different natural language processing techniques could further improve performance.

Speech Recognition Error Correction

We are in the process of applying similar techniques to speech recognition systems. This commonsense approach to predictive text entry can be used to improve error correction interfaces for such systems, as well as to disambiguate phonetically similar words and improve overall speech recognition accuracy.

ACKNOWLEDGMENTS

The authors would like to thank Push Singh and Hugo Liu for their helpful feedback and for breaking new ground with Open Mind Common Sense and OMCSNet. Thanks also to Kevin Brooks and Angela Chang at the Motorola Advanced Concepts Group and Paul Wisner and Franklin Reynolds at the Nokia Research Center for generously contributing their expertise and phones for our various implementations.

REFERENCES

1. Conrad, R., and Longman, D.J.A. Standard Typewriter versus Chord Keyboard: An Experimental Comparison. *Ergonomics*, 8 (1965), 77-88.
2. Darragh, J.J., Witten I.H., James, M.L. The Reactive Keyboard: A Predictive Typing Aid. *IEEE Computer* (23)11 (1990), 41-49.
3. Eagle, N., Singh, P., Pentland, A. Common Sense Conversations: Understanding Casual Conversation Using a Common Sense Database. *Proc. AI2IA Workshop at IJCAI 2003*.
4. Karat, C.-M., Halverson, C., Horn, D., and Karat, J. Patterns of Entry and Correction in Large Vocabulary Continuous Speech Recognition Systems. *Proc. CHI 2002*, 568-575.
5. Liu, H. Unpacking Meaning from Words: A Context-Centered Approach to Computational Lexicon Design. *Proc. CONTEXT 2003*, 218-232.
6. Liu, H. and Singh, P. OMCSNet: A Practical Commonsense Reasoning Toolkit. MIT Media Lab Society Of Mind Group Technical Report SOM02-01 (2002).
7. Silfverberg, M., MacKenzie, I.S., and Korhonen, P. Predicting Text Entry Speed on Mobile Phones. *Proc. CHI 2000*, CHI Letters 2(1), 9-16.
8. Singh, P. The Public Acquisition of Commonsense Knowledge. *Proc. 2002 AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*, 47-52.
9. Witten, I.H., Moffat, A., and Bell, T.C. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann, San Francisco, CA, 1999.
10. Zeno, S., et al. *The Educator's Word Frequency Guide*. Touchstone Applied Science Associates, 1995.
11. Zhai, S. and Kristensson, P.-O. Shorthand Writing on Stylus Keyboard. *Proc. CHI 2003*, CHI Letters 5(1), 97-104.