How to Wreck a Nice Beach You Sing Calm Incense

Henry Lieberman, Alexander Faaborg, Waseem Daher, José Espinosa
MIT Media Laboratory
20 Ames St., Bldg E15
Cambridge, MA 02139 USA
lieber@media.mit.edu, faaborg@media.mit.edu, wdaher@mit.edu, jhe@media.mit.edu

ABSTRACT
A principal problem in speech recognition is distinguishing between words and phrases that sound similar but have different meanings. Speech recognition programs produce a list of weighted candidate hypotheses for a given audio segment, and choose the "best" candidate. If the choice is incorrect, the user must invoke a correction interface that displays a list of the hypotheses and choose the desired one. The correction interface is time-consuming, and accounts for much of the frustration of today's dictation systems. Conventional dictation systems prioritize hypotheses based on language models derived from statistical techniques such as n-grams and Hidden Markov Models.

We propose a supplementary method for ordering hypotheses based on Commonsense Knowledge. We filter acoustical and word-frequency hypotheses by testing their plausibility with a semantic network derived from 700,000 statements about everyday life. This often filters out possibilities that "don't make sense" from the user's viewpoint, and leads to improved recognition. Reducing the hypothesis space in this way also makes possible streamlined correction interfaces that improve the overall throughput of dictation systems.

Author Keywords
Predictive Interfaces, Open Mind Common Sense, Speech Recognition.

ACM Classification Keywords
H.5.2 User Interfaces: *input strategies*.
I.2.7 Natural Language Processing: *Speech recognition and synthesis*

INTRODUCTION
Errors in text generated by speech recognition are usually easy for people to spot because they don't make sense. Even a well trained speech recognition system will occasionally produce a nonsensical (but phonetically similar) word or phrase. For instance, the phrases "recognize speech using common sense" and "wreck a nice beach you sing calm incense" while phonetically similar, are contextually very different. Because of this, acoustic analysis alone is not enough to accurately recognize speech. Speech recognition systems must also take into account the context of what the user is saying. Previous approaches to this problem have relied only on statistical techniques, calculating the probability of words appearing in a particular order.

We propose a new solution, using Commonsense Knowledge to understand the context of what a user is saying. A speech recognition system augmented with Commonsense Knowledge can spot its own nonsensical errors, and proactively correct them. Previously this approach has been successfully applied to similar problem of predictive text entry [12].

We have found that by filtering out hypotheses that don't make sense, we can improve overall recognition accuracy, and improve error correction user interfaces.

Previous Work
Previous approaches, surveyed by Jelinek, have used statistical language models, based on such techniques as Hidden Markov Models, and n-grams [1,2,3]. These models calculate the probability of each word in a vocabulary appearing next, based on the previous sequence of words. Kuhn adapted these models to weigh recently spoken words higher, improving accuracy [4]. He found that a recently spoken word was more likely to appear then either its overall frequency in the language or a Markov Model would suggest. Even with the best possible language models, these methods are limited by their ability to represent language statistically. In contrast, we propose using Commonsense Knowledge to solve the context problem with semantics in addition to statistics.

Open Mind: Teaching Computers the Stuff we All Know
Since the fall of 2000 the MIT Media Lab has been collecting commonsense facts from the general public through a Web site called Open Mind [5,6]. At the time of this writing, the Open Mind Common Sense Project has collected over 700,000 facts from over 14,000 participants. These facts are submitted by users as natural language statements of the form "*tennis is a sport*" and "*playing tennis requires a tennis racket.*" While Open Mind does not contain a complete set of all the common sense facts

found in the world, its knowledge base is sufficiently large enough to be useful in real world applications.

Using natural language processing, the Open Mind knowledge base was mined to create ConceptNet [7], a large-scale semantic network currently containing over 300,000 nodes. ConceptNet consists of machine-readable logical predicates of the form: [IsA "tennis" "sport"] and [EventForGoalEvent "play tennis" "have racket"]. ConceptNet is similar to WordNet [8] in that it is a large semantic network of concepts, however ConceptNet contains everyday knowledge about the world, while WordNet follows a more formal and taxonomic structure. For instance, WordNet would identify a *dog* as a type of *canine*, which is a type of *carnivore*, which is a kind of *placental mammal*. ConceptNet identifies a *dog* as a type of *pet* [7].

ConceptNet allows software applications to understand the relationships between concepts in thousands of domains. And this domain knowledge can be leveraged by speech recognition engines to understand that "I bought my dog in a pet shop" makes more sense then the phonetically similar phrase "I bought my dog in a sweatshop."

Re-Ranking the Candidate Hypotheses List

Our implementation accesses the Microsoft Speech Engine using the Microsoft Speech SDK 5.1. The application retrieves the Microsoft Speech engine's hypotheses for each speech utterance, and re-ranks the list based on the semantic context of what the user has previously said, using ConceptNet. Hypotheses that appear in the concepts context are moved toward the top of the list.

For instance, if the user says "my bike has a squeaky brake." The Microsoft Speech Engine often predicts the final word to be "break," (as in "to break something"). However, by using ConceptNet to understand the context of *bike* (which includes concepts like: *tire*, *seat*, *wheel*, *pedal*, *chain*… and *brake*), our application is able to correctly guess that the user meant the physical "brake". This is shown in Figure 1.

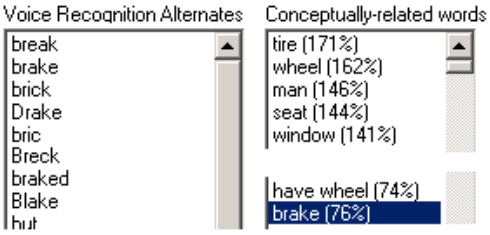

Figure 1: Re-Ranking the Candidate Hypotheses List

This example disambiguates between two words that are phonetically identical, ("break" and "brake"), but the approach also works well for disambiguating words that are phonetically similar (like "earned" and "learned").

EVALUATION

In a preliminary test, 3 subjects completed a 286-word dictation task. Their error correction interface contained a list of alternative hypotheses, and an "undo" button, which allowed them to dictate a phrase again. Each subject trained the speech recognition engine before running the experiment. After one initial training session, subjects were allowed to familiarize themselves with the error correction user interface. All subjects dictated the same text. We logged the ranking of alternative hypotheses for each speech utterance, the number of times the subject clicked the "undo" button, and the dictation time. After the subjects completed the dictation task, we analyzed their error logs to see if the augmented speech engine would have prevented the mistakes from occurring. We analyzed the error logs instead of directly testing our augmented speech recognition engine to control for the variability of acoustic input. We found that using Commonsense Reasoning to re-rank the candidate hypotheses would have prevented 17% of the errors from occurring, which would have reduced overall dictation time by 7.5%.

Additionally, we found that when subjects used the error correction interface, if the correct hypothesis appeared toward the bottom of the list they would often click the "undo" button and say the phrase again instead of reading the entire list. Because of this, we have found the augmented speech engine to be more efficient. Even if the first hypothesis is still incorrect, re-ranking the candidate hypotheses improves the error correction interface. Commonsense Reasoning helps speech recognition engines make better mistakes.

DISCUSSION

Previous techniques have focused on low-order word n-grams, where the probability of a word appearing is based on the n-1 words preceding it. The first difference between our approach and n-grams is the corpus being used. In our case the corpus consists of Commonsense Knowledge. While it is certainly possible to use an n-grams approach on a training corpus of Commonsense Knowledge, there are also many differences in how these two approaches function. The n-grams approach is a statistical technique based on frequency of adjacent words. Because n is usually 2 or 3 (referred to as bigrams and trigrams), this approach cannot take into account the context of what the user is saying beyond a three word horizon. To take into account more context, n would need to be increased to a larger number like 10. However, this results in intractable memory requirements. Given a training corpus containing m words, the n-grams approach where n equals 10 would require storing a lookup table with (in the worst case) m^{10} entries. The n-gram approach is usually trained on a corpus where m is in the millions. Commonsense Reasoning is able to escape these intractable memory requirements because (1) our training corpus is smaller, and (2) our parser dose more natural language processing. To look at an example, consider determining the last word in this sentence: "buy me a ticket to the movie, I'll meet you at the

(thée ətər)". The Commonsense Reasoning approach notices the words *ticket* and *movie*, and uses ConceptNet to look up the context of these two words (which returns list of about 20 words), it then concludes "theater" is the best guess. An n-grams approach would need to look up every other instance of this sentence before it could form its list of candidate hypotheses.

In addition to being more efficient, processing speech based on semantics may also be closer to how the human brain completes the task. Acero, Wang and Wang note that "Shortly after a toddler is taught that "dove" is a bird, she has no problem in using the word dove properly in many contexts that she hasn't heard before; yet training n-gram models require seeing all those n-grams before" [10]. The n-gram model, unlike semantics, does not generalize.

Improving Speech Recognition User Interfaces

Current speech recognition error correction interfaces require the user to read a list of candidate hypotheses and then make corrections by saying "pick n" (where n is the number of the correct word). The error correction interface for IBM's ViaVoice [11] is shown in Figure 2. Reading this list of hypotheses is time intensive, and slows down the rate at which users can dictate text.

Figure 2: IBM ViaVoice's Error Correction Interface

By using Commonsense Reasoning to order this list, the correct hypothesis is more likely to appear in the first several options. This allows the user to simply say "oops" and the incorrect word is replaced by the next hypothesis. Since the user's attention does not have to shift to the error correction window, they will not have to perform a visual search of the candidate hypotheses. This increases the overall rate at which users can dictate text, and streamlines the interface.

CONCLUSION

By using ConceptNet, a vast semantic network of Commonsense Knowledge, we are able to reduce the number of nonsensical errors produced by speech recognition engines. This increases overall accuracy, and can be used to streamline speech recognition error correction interfaces, saving user's time.

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 ConceptNet.

REFERENCES

1. Jelinek, F., Mercer, R.L., Bahl, L.R. A Maximum Likelihood Approach to Continuous Speech Recognition. IEEE Journal of Pattern Analysis and Machine Intelligence (1983); and Readings in Speech Recognition, A. Waibel, K.F. Lee, (eds.) pages 308-319. Morgan Kaufmann Publishers, San Mateo, CA. (1990).
2. Jelinek, F. The Development of an Experimental Discrete Dictation Recognizer. Proc. IEEE, Vol. 73, No. 11, pages 1616-1624. (1985).
3. Jelinek, F. Training and Search Models for Speech Recognition. Voice Communication between Humans and Machines. Roe, D. Wilpon, J. (eds.) Pages 199-214. Washington D.C.: National Academy Press. (1994).
4. Kuhn, R. Speech Recognition and the Frequency of Recently Used Words: a Modified Markov Model for Natural Language. Proceedings of the 12th conference on Computational linguistics - Volume 1. pages 348-350. Budapest, Hungry. (1988)
5. Singh, P. The Open Mind Common Sense project. KurzweilAI.net: http://www.kurzweilai.net/meme/frame.html?main=/articles/art0371.html (2002)
6. 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.
7. Liu, H. and Singh, P. ConceptNet: A Practical Commonsense Reasoning Toolkit. BT Technology Journal. Kluwer. (2004)
8. Fellbaum, C. WordNet: An Electronic Lexical Database. MIT Press, Cambridge, MA, USA, (1998)
9. Microsoft Speech SDK 5.1 for Windows Applications: http://www.microsoft.com/speech/download/sdk51/
10. Acero, A., Wang, Y., and Wang, K. A Semantically Structured Language Model, in Special Workshop in Maui (SWIM), Jan 2004
11. IBM ViaVoice: http://www.scansoft.com/viavoice
12. Stocky, T. Faaborg, A. Lieberman, H. A Commonsense Approach to Predictive Text Entry, Conference on Human Factors in Computing Systems CHI2004, Vienna, Austria. (2004)