Spoken Language Technology Research at Microsoft

X.D. Huang

Speech Technology Group, Microsoft Research, Redmond, WA 98052, USA

Abstract: The Microsoft Speech Technology Group engages in research and development of spoken language technologies for use by a broad audience of computer users and in a wide range of applications. We are researching how to voice-enable the whole computer system, and not just a portion of it. Our current work in support of this broad objective falls into four areas: speech recognition technology code-named Whisper; speech synthesis technology code-named Whis
ters; spoken language understanding technology code-named Leap; and the Speech Software Developer's Kit that facilitates full integration of speech into applications via Microsoft Speech Application Programming Interface (SAPI).

SPEECH RECOGNITION RESEARCH

Microsoft's Whisper engine offers general purpose speaker-independent continuous speech recognition (CSR). Whisper can be used for command and control, dictation and conversational applications. Whisper offers many features such as continuous speech recognition; speaker-independence with adaptation; and dynamic vocabulary. Whisper has a unified architecture that can be scaled to meet different application and platform requirements.

On the technology side, the Whisper system contributes innovations in the areas of both acoustic modeling and language modeling. These technology enhancements include normalized feature representations for improved robustness, senone models derived from inter- and intra-word context-dependent phonemes, semi-continuous or continuous density hidden Markov models implemented with the generic shared density function architecture, and efficient decoder algorithms. Our goal is to develop a general purpose speaker-independent continuous speech recognition (CSR) engine that can recognize unrestricted text and is effective for command and control, dictation and conversational systems. In pursuing this goal, we recognize that speakers with different backgrounds, expectations, fluencies, and dictation purposes may speak with varying styles and rates, and neither style (isolated or continuous) should be penalized for the sake of the other. We recently benchmarked Whisper with the widely used DARPA SI-284 training set and the 1994 H1 development test set. Whisper's word recognition error rate was 7.7% using the trigram for the H1 DEV94 test set. A simplified version of Whisper can be downloaded from http://research.microsoft.com/stg.

To further improve the performance of Whisper, we are experimenting with unified acoustic and language models, dynamically configurable and adaptive systems, improved noise immunity, and integrated semantic processing.

SPEECH SYNTHESIS RESEARCH

Microsoft's Whistler Text to speech (TTS) system was designed so that we can automatically construct the model parameters from training data. Whistler can produce synthetic speech that reasonably resembles the acoustic and prosodic characteristics of the original speaker. The underlying technologies used in Whistler can significantly facilitate the process of creating generic TTS systems for a new language, a new voice, or a new speech style. A simplified version of Whistler can be downloaded from http://research.microsoft.com/stg. The memory footprint of this version is about 3MB.

Our preliminary work indicated that we could effectively leverage speech recognition technology to significantly improve the naturalness of TTS systems. Whistler benefited substantially from stochastic learning techniques that have been widely used in speech recognition. The data-driven approach could help to create speech output that has a paradigm-shift impact. We think that factors such as speaking style, utterance situation and the speaker's mental states could all be modeled in Whistler's stochastic data-driven framework in the future. For ongoing research to further improve Whistler, we are experimenting with our Natural Language Understanding System to improve our text analysis and prosody models, longer units for difficult contexts (like vowel-vowel transitions) and/or frequent contexts (like the most frequent triphones or words), and improved pitch tracking and vocal modeling.

SPOKEN LANGUAGE UNDERSTANDING RESEARCH

Our Leap Project's main goal is to provide Spoken Language Understanding (SLU) abilities to applications in the form of an SDK. This allows people to express themselves using whatever words they naturally use to have...
computers intelligently understand their purpose. Leap will not only provide necessary SLU components to support parsing and conversation management but also needed tools for developers to author whatever applications they are interested in developing.

Leap used a similar design philosophy of Carnegie Mellon University's Phoenix for developing simple but robust natural language interfaces to applications. It uses frames to represent semantic relations. A frame represents some basic type of action for the application. Slots in a frame represent the information that is relevant to the action. We are extending the Phoenix architecture with hierarchical frame matching to simplify authoring, with a portable conversation component that is driven by application-specific semantic grammars and dialog specification, and with multi-application COM-based architecture. We are also experimenting with our Natural Language Understanding System to simplify authoring of the semantic grammar.

**SPEECH SDK AND SPEECH API DEVELOPMENT**

The Microsoft Speech SDK 4.0 specifies and enables the Microsoft Speech Application Programming Interface (SAPI), which is the framework for integrating speech into applications. It enables development for command and control speech recognition, dictation, and text-to-speech, in C, C++, Visual Basic, or Visual J++. Microsoft is committed to making SAPI accessible from the widest variety of languages possible, and we're working to make that even easier in the future. Writing a speech-enabled application requires a developer to have the SAPI interfaces that come in the SDK, as well as one or more "engines" which implement the desired functionality, e.g. speech synthesis and/or "Command and Control" speech recognition.

Developers can acquire an engine from a number of different sources, and each engine has its own advantages, disadvantages and specialties. Microsoft encourages developers to take the time to evaluate different speech vendors' offerings to find one that best suits their particular needs. The SDK download package includes: API documentation, sample applications, sample source code, and redistributable API files. Microsoft Speech SDK can be downloaded from http://research.microsoft.com/stg. We will incorporate Leap as well as improved Whisper and Whistler in our future SDK release.

**SUMMARY**

One overall theme of our unified program is technology and component sharing for all speech research and development tasks and systems. Areas where significant tool and component sharing across speech recognition, spoken language understanding, and text-to-speech are in place or anticipated include: text normalization; orthography-to-phoneme conversion context management; semantic networks and parsing engines; speech unit segmentation; speech analysis and coding; prosodic feature analysis; and analysis and annotation of corpora.

To develop successful conversational applications, we need to make it easier for others to create new applications. This means that not only must we make Whisper, Whistler, and Leap more robust for new tasks but also continue to enhance the tools and sample SAPI applications in the Speech SDK so that necessary customization work can be done easily.

**REFERENCES**