Editor: Christof Ebert
Vector Consulting Services
christof.ebert@vector.com

# Speech Recognition for Voice-Based Machine Translation

Tiago Duarte, Rafael Prikladnicki, Fabio Calefato, and Filippo Lanubile

Douglas Adams described a mythical Babel fish: "If you stick one in your ear, you can instantly understand anything said to you in any form of language." We aren't there yet, but real-time voice-based machine translation is quickly progressing. It is stimulated by many international teams who want to understand each other syntactically as well as semantically. Authors Tiago Duarte, Rafael Prikladnicki, Fabio Calefato, and Filippo Lanubile provide an overview of current technologies for real-time voice-base machine translation. I look forward to hearing from both readers and prospective column authors about this column and the technologies you want to know more about. —*Christof Ebert*

**MACHINE TRANSLATION (MT)** is a subfield of computational linguistics that investigates the use of software to translate text or speech from one natural language to another.[1,2] It can be especially useful for performing tasks that involve understanding and speaking with people who don't speak the same language. In the global software engineering (GSE) domain, for example, language is an important factor in the success of offshore IT work in countries with strong English language capabilities, such as Ireland, the Philippines, India, and Singapore.[2]

Several countries are trying to increase their presence in today's global IT market, but they lack English-speaking professionals.[3] For this reason, distributed project meetings, such as requirements workshops, can benefit from MT to help bridge the communication gap.

## Background on Machine Translation

The idea of using digital computers to translate natural languages first emerged about 50 years ago.[2] The communication technology available today—specifically, anything that enables real-time, online conversation—is getting a tremendous amount of attention, mostly due to the Internet's continuous expansion. The rise of social networking has also contributed to this growing interest as more users join and speak different languages to communicate with each other. But in spite of this technology's recent progress, we still lack a thorough understanding of how real-time MT affects communication.[1]

MT is challenging because translation requires a huge amount of human knowledge to be encoded in machine-processable form. In addition, natural languages are highly ambiguous: two languages seldom express the same content in the

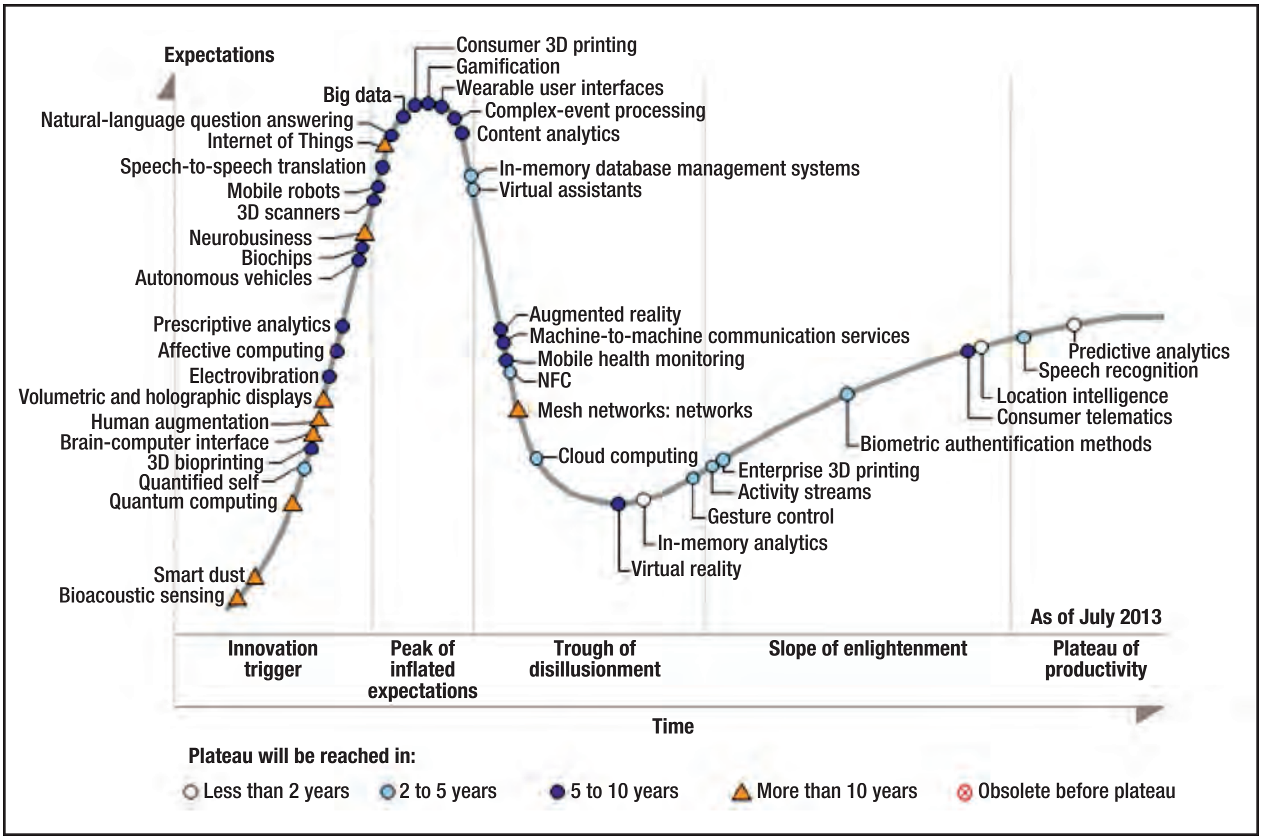


**FIGURE 1.** The Gartner Hype Cycle for emerging technologies in 2013 (www.gartner.com/newsroom/id/2575515).

same way.[4] Google Translate is an example of a text-based MT system that applies statistical learning techniques to build language and translation models from a large number of texts. Other tools offer cross-language chat services, such as IBM Lotus Translation Services for Sametime and VoxOx.

In the 2013 version of the Gartner Hype Cycle for emerging technologies (see Figure 1), we can see an important trend in speech-to-speech recognition technology (the translation of spoken words into text and back to speech in the translated language), which leads us to predict that in a few years' time, we might see speech-to-speech or voice-based MT technology.

Although speech-to-speech translation technology is considered to be an innovation trigger—there's a potential technology breakthrough but no usable products or proven commercial viability—speech recognition technology falls in the plateau of pro-

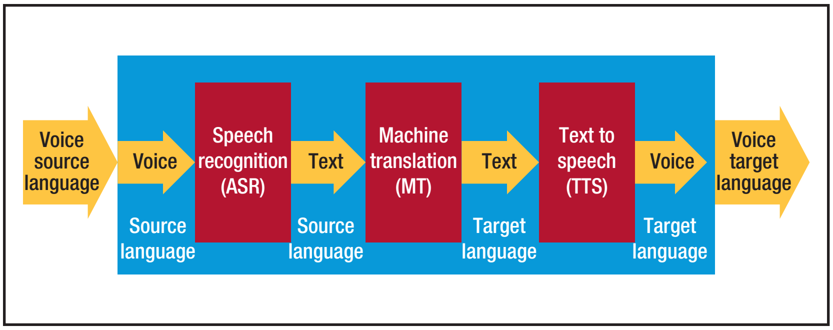


**FIGURE 2.** Machine translation (MT) components. The automatic speech recognition component processes the voice in its original language, resulting in a text version that goes through the MT component, which translates it to the target language. At the end of the process, this translated text goes through the text-to-speech component, which "speaks" the text in a synthesized voice.

ductivity, meaning that mainstream adoption is starting to take off.

Speech-to-speech translation has three components[5]: automatic speech recognition (ASR), MT, and voice synthesis (or text to speech; TTS). As shown in Figure 2, the ASR component processes the voice in its original language, creating a text version of what the speaker said. This text in the original language goes through the MT component, which translates it to the target language. Finally, this translated text goes through the TTS component, which "speaks" the text using a synthesized voice in the target language. For each step of this process, many other technologies could be used in the future to improve the quality of the overall speech-to-speech translation.

## Available Technology for Speech Recognition

As part of a program of research on speech-to-speech translation, we review some of the available technologies for speech recognition, the first component in any voice-based MT system (see Table 1).

### Microsoft Speech API

Microsoft Speech API (SAPI) allows access to Windows' built-in speech recognition and speech synthesis components. The API was released as part of the OS from Windows 98 forward. The most recent release, Microsoft Speech API 5.4, supports a small number of languages: American English, British English, Spanish, French, German, simplified Chinese, and traditional Chinese. Because it is a native Windows API, SAPI isn't easy to use unless you're an experienced C++ developer (http://msdn.microsoft.com/en-us/library/hh323805(v=office.14).aspx).

The Microsoft .NET framework offers an alternative way to access Windows' speech resources through the System.Speech namespace. This library gives C# developers access to the same SAPI features through much simpler interfaces. Its speech recognition components allow the system to interpret strings from single spoken words up to complete phrases; typically, however, developers use Microsoft speech technologies to let applications recognize spoken, predefined commands instead of complex phrases. In these cases, the accuracy of the speech recognition is very high.

### Microsoft Server-Related Technologies

The Microsoft Speech Platform provides access to speech recognition and synthesis components that encourage the development of complex voice/telephony server applications. This technology supports 26 different languages, although it primarily just recognizes isolated words stored in a predefined grammar (http://msdn.microsoft.com/en-us/library/hh361571(v=office.14).aspx).

Microsoft also provides the Microsoft Unified Communications API (UCMA 3.0), a target for server application development that requires integration with technologies such as voice over IP, instant messages, voice call, or video call. The UCMA API allows easy integration with Microsoft Lync and enables developers to create middle-layer applications.

### Sphinx

Sphinx 4 is a modern open source speech recognition framework based on hidden Markov models (HMMs) and developed in the Java programming language. This free platform also allows the implementation of continuous-speech, speaker-independent, and large-vocabulary recognition systems (http://cmusphinx.sourceforge.net/sphinx4).

The framework is language independent, so developers can use it to build a system that recognizes any language. However, Sphinx requires a model for the language it needs to recognize. The Sphinx group has made available models for English, Chinese, French, Spanish, German, and Russian languages (http://cmusphinx.sourceforge.net/wiki/faq).

TABLE 1

Technologies for speech recognition.

| Technologies | Type | Recognition type | Vocabulary size allowed | Common usage | Recognition quality | Default support languages | New languages allowed | Speech synthesis supported | Free or open source |
|---|---|---|---|---|---|---|---|---|---|
| Microsoft Speech API | Windows COM API | Dictation, complex phrase recognition | Large | Desktop application development | High | Few | No | Yes | No |
| Microsoft .NET System. Speech namespace | Windows .NET API | Dictation, complex phrase recognition | Large | Desktop application development | High | Few | No | Yes | No |
| Microsoft Speech Platform | Windows API | Commands, isolated word recognition | Large | Server application development | n/a | Many | No | Yes | No |
| Microsoft Unified Communications API | Windows API | Commands, isolated word recognition | Large | Server application development | n/a | Many | No | Yes | No |
| Sphinx 4 | Complete Framework for SR | Dictation, complex phrase recognition | Large; depends on implementation | Speech recognition research studies | Depends on implementation | Few | Yes | No | Yes |
| HTK | Complete Framework for SR | Dictation, complex phrase recognition | Large; depends on implementation | Speech recognition research studies | Depends on implementation | None | Yes | No | No; license might be needed |
| Julius | Decoder system | Dictation, complex phrase recognition | Large | Speech recognition research studies | Depends on implementation | None | No | No | Yes |
| Java Speech API | Specification | n/a | n/a | n/a | Depends on implementation | None | No | Depends on implementation | Yes |
| Google Web Speech API | JavaScript API for Chrome | Dictation, complex phrase recognition | Large | Web application development | Average | Many | No | Not yet | Yes |
| Nuance Dragon SDK | API for desktop and mobile application | Dictation, complex phrase recognition | Large | Client, server and mobile application development | High | Many | No | Yes | No |

Sphinx 4 is a flexible, modular, pluggable framework that is fostering a lot of the current research in speech recognition; many studies use it to create speech recognition systems for new languages or testing algorithms (http://cmusphinx.sourceforge.net/sphinx4/doc/Sphinx4Whitepaper.pdf).

Mainstream adoption of machine translation is starting to take off in industry.

### HTK

The HMM Toolkit, also known as HTK, is a platform for building and manipulating HMMs. Researchers primarily use HTK for speech recognition, although it has been used for other applications such as research about speech synthesis, character recognition, and DNA sequencing. HTK can build complete continuous-speech, speaker-independent, and large-vocabulary recognition systems for any desired language. It also provides tools to create and train acoustic models (http://htk.eng.cam.ac.uk/docs/faq.shtml).

Microsoft bought the platform in 1999 and retains the copyright to existing HTK code. Developers can still use HTK to train the models used in their products, but the HDecoder module has a more restrictive license and can be used only for research purposes. (The HDecoder is the module responsible for analyzing the digital signal and identifying which is the most likely word or phrase being said according to the acoustic and language models available.)

### Julius

Julius is a high-performance decoder designed to support a large vocabulary and continuous speech recognition, which are important features to allow the implementation of dictating systems. The decoder is at the heart of a speech recognition system; its job is to identify the most likely spoken words for given acoustic evidence. It can perform near-real-time decoding for 60,000-word dictation tasks on most current PCs. This decoder implements the most popular speech recognition algorithms, which makes it very efficient. The system also allows the usage of models created in different tools such as HTK and the Cambridge Statistical Language Modeling toolkit (CAM SLM) from Carnegie Mellon University (CMU; http://julius.sourceforge.jp/en_index.php).

A great advantage of using the Julius decoder is that it's freely available. The VoxForge project is currently developing acoustic models to be used in the Julius recognition system. The main platform for Julius is Linux, although it works on Windows as well; it also has a Microsoft SAPI–compatible version.

### Java Speech API

The Java Speech API (JSAPI) is a specification for cross-platform APIs that supports command-and-control recognizers, dictation systems, and speech synthesizers. Currently, the Java Speech API includes javadoc-style API documentation for the approximately 70 classes in and interfaces to the API. The specification includes a detailed programmer's guide that explains both introductory and advanced speech application programming with JSAPI, but it doesn't yet offer the source code or binary classes required to compile the applications (www.oracle.com/technetwork/java/jsapifaq-135248.html).

JSAPI is freely available, and its owners welcome anyone to develop an implementation for it; so far, it has just a few implementations, such as FreeTTS for voice synthesis and IBM Speech for Java for speech recognition (the discontinued IBM ViaVoice).

### Google Web Speech API

In early 2013, Google released Chrome version 25, which included support for speech recognition in several different languages via the Web Speech API. This new API is a JavaScript library that lets developers easily integrate sophisticated continuous speech recognition feature such as voice dictation in their Web applications. However, the features built using this technology can only be used in the Chrome browner; other browsers don't support the same JavaScript library (http://chrome.blogspot.com.br/2013/02/bringing-voice-recognition-to-web.html).

### Nuance Dragon SDK

Dragon Naturally Speaking by Nuance Communications is an application suite for speech recognition, supporting several languages other than English, including French, German, Italian, and Dutch. It's available as a desktop application for

PC and Mac and as a mobile app for Android and iOS. Nuance also provides software development kits (SDKs) for enabling speech recognition in third-party applications. Developers use the Dragon SDK client to add speech recognition to existing Windows applications, the SDK as a back end to support non-Windows clients, and the Mobile SDK to develop apps for iOS, Android, and the Windows Phone.

MT adoption is starting to take off in industry. MT technology technology is currently available in the form of cross-language Web services that can be embedded into multiuser and multilingual chats without disrupting conversation flow, but it's mostly text-based. Many factors affect how MT systems are used and evaluated, including the intended use of the translation, the nature of the MT software, and the nature of the translation process. Voice-based technology is already available, but it isn't capable of handling typical human conversations where people talk over one another, use slang, or chat on noisy streets. Moreover, to overcome the language barrier worldwide, multiple languages must be supported by speech-to-speech translation technology, requiring speech, bilingual, and text corpora for each of the several thousands of languages that exist on our planet today. To achieve more sophistication and accuracy, research and development must be further accelerated in this area.

## Acknowledgments
This work is partially funded by the Rio Grande do Sul State funding agency (FAPERGS). Rafael Prikladnicki is a CNPq researcher (309000/2012-2).

## References
1. H.C. Wang, S. R. Fussel, and D. Cosley, "Machine Translation vs. Common Language: Effects on Idea Exchange in Cross-Lingual Groups," *Proc. Computer-Supported Cooperative Work* (CSCW 13), ACM, 2013, pp. 935–937.
2. F. Calefato, F. Lanubile, and R. Prikladnicki, "A Controlled Experiment on the Effects of Machine Translation in Multilingual Requirements Meetings," *Proc. 6th IEEE Int'l Conf. Global Software Eng.* (ICGSE 11), IEEE, 2011, pp. 94–102.
3. F. Calefato et al., "Assessing the Impact of Real-Time Machine Translation on Requirements Meetings: A Replicated Experiment," *Proc. ACM-IEEE Int'l Symp. Empirical Software Engineering and Measurement* (ESEM 12), ACM, 2012, pp. 251–260.
4. D. Arnold, "Why Translation Is Difficult for Computers," *Computers and Translation: A Translator's Guide*, H. Somers, ed., Benjamins Translation Library, 2003, pp. 119–142.
5. A. Waibel and C. Fugen, "Spoken Language Translation," *Signal Processing Magazine*, vol. 25, no. 3, 2008, pp. 70–79.

**TIAGO DUARTE** is a master's student in the Computer Science School at Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS). He's also a project manager with more than five years' experience with global projects and distributed teams within multinational organizations. Contact him at tiagoduarte@gmail.com.

**RAFAEL PRIKLADNICKI** is an associate professor in the Computer Science School and director of the Scientific and Technology Park (TECNOPUC) at Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), where he also leads the MuNDDoS research group (www.inf.pucrs.br/munddos), studying topics such as global software engineering, agile methodologies, crowdsourcing, and high-performance software development teams. Contact him at rafaelp@pucrs.br or via www.inf.pucrs.br/~rafael.

**FABIO CALEFATO** is a postdoc at the University of Bari, where he received a PhD in computer science. His research interests include computer-mediated communication and global software engineering. Contact him at fabio.calefato@uniba.it.

**FILIPPO LANUBILE** is an associate professor of computer science at the University of Bari, where he leads the Collaborative Development Group (http://cdg.di.uniba.it). His research interests include collaborative/social software engineering and global software development. Contact him at filippo.lanubile@uniba.it.