PodCastle: A Spoken Document Retrieval Service Improved by Anonymous User Contributions

Masataka Goto and Jun Ogata

National Institute of Advanced Industrial Science and Technology (AIST)
1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, JAPAN
m.goto [at] aist.go.jp

Abstract. In this invited paper, we introduce a public web service, *PodCastle*, that provides full-text searching of speech data (Japanese podcasts) on the basis of automatic speech recognition technologies. This is an instance of our research approach, *Speech Recognition Research 2.0*, which is aimed at providing users with a web service based on Web 2.0 so that they can experience state-of-the-art speech recognition performance, and at promoting speech recognition technologies in cooperation with anonymous users. PodCastle enables users to find podcasts that include a search term, read full texts of their recognition results, and easily correct recognition errors by simply selecting from a list of candidates. Even if a state-of-the-art speech recognizer is used to recognize podcasts on the web, a number of errors will naturally occur. PodCastle therefore encourages users to cooperate by correcting these errors so that those podcasts can be searched more reliably. Furthermore, using the resulting corrections to train the speech recognizer, it implements a mechanism whereby the speech recognition performance is gradually improved. In our experiences from its practical use over the past 46 months (since December, 2006), we confirmed that the performance of PodCastle was improved by a number of anonymous user contributions.

Keywords: information retrieval, speech recognition, error correction, wisdom of crowds, Web 2.0

1 Introduction

Speech recognition researchers understand what sort of speech is easily recognized by speech recognizers and realize that speech recognizers perform best when dealing with clean speech. On the other hand, most end users of speech recognizers judge the effectiveness of speech recognition from their limited experiences and do not necessarily understand how useful state-of-the-art recognizers can be. Users sometimes do not adequately comprehend what sort of voices or recording conditions make recognition difficult. If they have previously had difficulty being understood by speech recognizers, they often doubt the usefulness of speech recognition and may stop using it.

The first aim of this study is to address this problem by promoting the popularization and use of speech recognition by raising end user awareness of state-of-the-art speech recognition performance. For this purpose, we launched a podcast search web service called *PodCastle* (Goto *et al.*, 2007; Ogata *et al.*, 2007; Ogata and Goto, 2009b; Ogata and Goto, 2009a) in 2006 that allows anonymous users to search and read *podcasts*, and to share the full text of speech recognition results for podcasts. Podcasts are audio programs distributed on the web, like radio shows or audio blogs. They are becoming increasingly popular because updated podcasts (MP3 audio files) can be easily and frequently downloaded by using RSS syndication feeds. Since various contents have already been published as podcasts, users can grasp the current state of speech recognition technology just by seeing the results of speech recognition applied to published podcasts. This is

important because when some users experience recognition errors while speaking into a microphone, they may become uncomfortable or frustrated and lose their motivation. Such problems do not occur for PodCastle because users do not have to provide their own speech input at all.

However, even state-of-the-art speech recognizers cannot correctly transcribe all podcasts, because their contents and recording environments vary very widely. A typical approach to deal with speech contents that cannot be properly recognized is to create a speech corpus including such contents and prepare correct transcriptions to train speech recognizers. This approach, however, is impractical for PodCastle because advance preparation of a corpus covering diverse podcast contents will be too costly and time consuming.

The second aim of this study is to dispense with the idea of using a pre-prepared corpus to address this problem, and instead employ the efforts of a large number of users to improve speech recognition and full-text search performance. Even if a state-of-the-art speech recognizer is used to recognize podcasts on the web, a number of errors will naturally occur. PodCastle therefore encourages users to cooperate by correcting these errors so that those podcasts can be searched more reliably. Furthermore, using the resulting corrections to train the speech recognizer, it implements a mechanism whereby the speech recognition performance is gradually improved. This approach can be described as *collaborative training for speech recognition*.

In 2006, we coined the term *Speech Recognition Research 2.0* (Goto *et al.*, 2007) to refer to the research approach where the current state of speech recognition technology is intentionally disclosed to users so that speech recognition performance can be improved through cooperative participation by users. This term was chosen to reflect the concept of Web 2.0 (O'Reilly), since this approach brings the benefits of Web 2.0 to speech recognition research. In Section 2 of this paper, we discuss the research approach that Speech Recognition Research 2.0 represents, and in Section 3 we describe the PodCastle web service as an instance of this approach. In Section 4, we summarize the contributions of this research.

2 Speech Recognition Research 2.0

Speech Recognition Research 2.0 is a new research approach to speech recognition which aims at improving speech recognition performance and the usage rate while benefiting from the cooperation of a number of anonymous end users. To achieve this, we proposed setting into motion a positive spiral as explained in Figure 1. In the past, this spiral has not necessarily taken hold because of inhibiting factors in each of the three steps important for popularizing speech recognition. The problems affecting each of these steps are as follows:

- With regard to (i) understanding speech recognition performance, users have tended to see the results of speech recognition applied only to their own voices. Once users experience recognition problems with their voices, they tend to incorrectly assume that other people's voices will also not be well recognized. On the other hand, speech recognition researchers have a better understanding of recognition capabilities because they have more opportunity to see the results of speech recognition applied not only to their own voices but to a large speech corpus.
- As for (ii) contributing to improved speech recognition performance, users of speech dictation systems can make speech recognizers adapt to their voices by reading out predefined sentences or add out-of-vocabulary words.[1] However, such in-house performance improvements made by each end user are not made available for re-use by other users. Only speech recognition researchers have been able to improve the performance of speech recognition as

[1] Some systems can adapt automatically to a speaker's voice during use without the user's awareness, and some systems can automatically acquire out-of-vocabulary words. However, in neither of these cases it is possible to share this information between users.

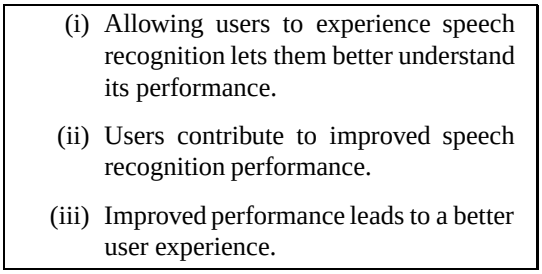

Figure 1: A positive spiral leading towards greater use of speech recognition (through repetition of steps (i), (ii), and (iii)).

Table 1: A comparison of the conventional approach to speech recognition research (Speech Recognition Research 1.0) and the proposed approach (Speech Recognition Research 2.0).

Speech Recognition Research 1.0	Speech Recognition Research 2.0
Stand-alone application	Web service
Dictation	Searching/browsing
Corpus	Web-based data
Limited topics	Unlimited topics
Transcription	Annotation
Out-of-vocabulary words	Not-yet-annotated words
Specialist participation	User participation
Individual correction	Social correction
Personal wisdom	Wisdom of crowds
Completed version	Perpetual beta

The above table is influenced by the comparison of Web 1.0 and Web 2.0 by O'Reilly (O'Reilly). Research projects that feature more points on the right side of the table are more worthy of the name Speech Recognition Research 2.0. *However, as with Web 2.0, this does not mean that all these points have to be featured in any one project.*

a whole. Consequently, it has been difficult to motivate a number of users to contribute to collaborative efforts that will improve performance.

- With regard to (iii) a better user experience, users have had little opportunity to experience the better performance that results from ongoing improvements made by researchers. For example, even if speech recognizers are made available as open source software (e.g., (Lee *et al.*, 2001)), these are mainly aimed at developers; end users have little opportunity to use them directly. Also, users of most speech recognition products have only experienced performance improvements through infrequent software updates.

By addressing these problems, Speech Recognition Research 2.0 aims to change the usage of speech recognition by setting the positive spiral of Figure 1 into motion. Table 1 compares this approach with the conventional approach to speech recognition research, which we will call *Speech Recognition Research 1.0*. We are not suggesting, though, that Speech Recognition Research 1.0 is inferior or obsolete, and there is no doubt that continued research using the Speech Recognition Research 1.0 approach is needed. We ourselves have continued our work on Speech Recognition Research 1.0 as the foundation for 2.0. It should also be stressed that we are discussing research approaches, and not speech recognition techniques or algorithms themselves, which is why we use the term *Speech Recognition Research 2.0* instead of *Speech Recognition 2.0*.

In the following, we discuss how the beneficial spiral of Figure 1 can be put into effect while also explaining the points in Table 1.

- Instead of developing stand-alone applications such as dictation or spoken dialogue by preparing a corpus for speech recognizers, Speech Recognition Research 2.0 provides a *web service* that allows users to *search and browse* open-to-the-public *web-based speech data* such as podcasts. In this way, it promotes understanding of speech recognition performance (step (i) in Figure 1).
- When recognizing web-based speech data, however, we cannot limit the range of topics and prepare in advance a suitable corpus with its transcription. This causes many recognition errors, so Speech Recognition Research 2.0 gets users to correct the errors, thereby enabling recognition of a wide variety of speech data on an *unlimited range of topics*. That is, users cooperate in the preparation of full-text transcriptions as a form of *annotation* that can be used when searching for speech data. It is important that user corrections are also used for training speech recognizers so that not-yet-corrected errors in other parts or other speech data can be reduced.[2] In Speech Recognition Research 2.0, out-of-vocabulary words are regarded as being nothing more than *not-yet-annotated words* which will be annotated (corrected) by users and then automatically added to the system vocabulary. In this way, users can contribute to improved performance (step (ii) in Figure 1).
- Furthermore, instead of confining this to individual corrections, we propose extending this *user participation* framework to provide a *social correction* framework, where a number of anonymous users can improve the performance by sharing their correction results over a web service. In this social framework, users gain a real sense of contributing to the convenience of other users, and can be motivated to contribute by seeing the correction activities made by other users. In this way, we can use the *wisdom of crowds* to achieve a better user experience (step (iii) in Figure 1).

In other words, Speech Recognition Research 2.0 can be described as an approach whereby a web service based on speech recognition that is permanently in beta version (*perpetual beta*) is launched and then improved by inviting users to use it on the web, thereby advancing the research.

As the first instance of Speech Recognition Research 2.0, we initiated the PodCastle project in January 2006. In this project, our goal is to set the positive spiral of Figure 1 into motion by providing the PodCastle web service which is based on the concepts of both Web 2.0 and Speech Recognition Research 2.0.

3 PodCastle: A Podcast Search Service Based on Speech Recognition

PodCastle is a social annotation web service where users can search, read, and annotate podcasts in text form. Each podcast consists of a series of episodes of audio data (MP3 files) and their metadata (RSS syndication feed) that promotes its circulation. The creator of a podcast (the podcaster) can add new episodes at arbitrary intervals (daily, weekly, etc.). With RSS, updated episodes are automatically downloaded from the web and can be stored in any type of player. Podcasts are often referred to as audio blogs and their popularity has grown because anybody can publish and download audio data with ease. Just as full-text search services are essential for accessing text web pages, there is a growing need for full-text speech retrieval services such as PodCastle.

Although there were previous research projects for speech retrieval (Whittaker *et al.*, 1999; Thong *et al.*, 2002; Lee and Chen, 2005) (Cambridge Multimedia Document Retrieval Project; CMU Informedia Digital Video Library Project) before 2006, most do not provide public web services for podcasts. There were two major exceptions, Podscope (Podscope) and PodZinger (PodZinger), which started web services for speech retrieval targeting English-language podcasts

[2] This is an original benefit of Speech Recognition Research 2.0 that is not provided by Web 2.0. For example, in other services such as Wikipedia (Wikipedia) based on the wisdom of crowds, the users' contributions are limited to the articles they edit. There is no automatic improvement of other articles.

in 2005. These services use speech recognition to turn podcasts into text, and can display a list of podcasts that include a search term. In Podscope, users are shown none of the speech recognition results and only the title list is provided, although speech data around the found search term can be played back. In PodZinger, users are shown text excerpts (recognition results) surrounding the search term, allowing users to grasp the context of the podcast more easily. In contrast, PodCastle is the first service to provide full-text searching of Japanese-language podcasts. Even if the other services can also support Japanese podcasts in the future, PodCastle differs significantly in three ways:

1. Although speech recognition has been used in earlier services, they have only displayed parts of the resulting text, making it impossible to visually ascertain the detailed contents of the podcast without actually listening to it.
2. The full-text results of speech recognition have been hidden inside, so it has not been possible to search them using other existing text-based search engines.
3. Even if users find that search results are degraded by unavoidable speech recognition errors, the users have had no means of correcting these errors.

In contrast, PodCastle allows full-text results of speech recognition to be accessed by both users and external search services, and allows a number of users to cooperate with each other to improve the speech recognition performance.

3.1 Three Functions of PodCastle

PodCastle supports three functions — searching, reading, and annotating — to satisfy all the points of Speech Recognition Research 2.0 listed in Table 1 and set the positive spiral of Figure 1 into motion. Specifically, the searching and reading functions let users better understand the speech recognition performance regarding podcasts (step (i) in Figure 1), and the annotating (error correction) function allows them to contribute to improved performance (step (ii)). This improved performance can then lead to a better user experience of searching and reading podcasts (step (iii)).

3.1.1 Searching Function This is a function that allows a full-text search of the speech recognition results (and the results corrected by users). When the user types in a search term, as with an ordinary text-based search engine, a list of episodes containing this term is displayed together with text excerpts of speech recognition results around the highlighted search term. These excerpts can be played back individually. By selecting one of these search results, the user is then able to access its full text by switching over to the reading function.

3.1.2 Reading Function With this function, as well as listening to a podcast the user can also view the text of the podcast. This allows users to understand the contents of a podcast even when audio playback is not possible, and allows them to quickly decide whether they are interested in the podcast's contents without having to listen to it. To make errors easy to discover, each word is colored according to the degree of reliability estimated during speech recognition. Furthermore, a cursor moves across the text in synchronization with the audio playback.

Because the full-text result of speech recognition being applied to each episode becomes available to external full-text search engines, such results can be discovered together with ordinary web pages by these engines. This increases the value of podcasts by bringing more users into contact with them. Since this benefits the podcasters, it will motivate them, together with other volunteer users, to use the annotating function.

3.1.3 Annotating Function This function allows users to add "annotations" to correct any recognition errors they may come across while searching or reading. Here, annotation means transcribing the podcast contents, either by selecting the correct candidate from the list of competitive

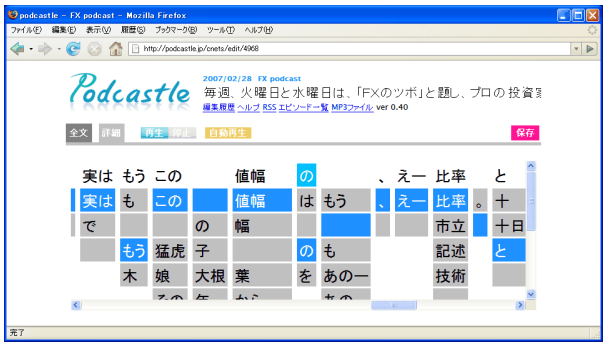

Figure 2: PodCastle screen snapshot of an interface for correcting speech recognition errors (competitive candidates are presented underneath the normal recognition results). Five errors in this excerpt were corrected by selecting from the candidates. The corrected Japanese sentence means *"... well, actually the ratio of this price range and ..."*.

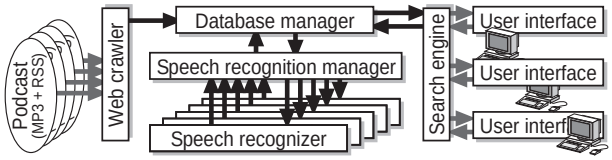

Figure 3: Implementation overview of PodCastle.

candidates, or by typing in the correct text. For this purpose, we provide an efficient error correction interface we earlier proposed (Ogata and Goto, 2005). In this interface, shown in Figure 2, a recognition result excerpt is shown around the cursor and scrolled in synchronization with the audio playback. Each word in the excerpt is accompanied by other word candidates, which are generated beforehand by using a *confusion network*[3] (Mangu *et al.*, 2000) that can condense a huge internal word graph of a large vocabulary continuous speech recognition (LVCSR) system. Note that users are not expected to correct all the errors, but can be expected to correct some errors according to their interests.

3.2 Implementation of PodCastle

The implementation overview of PodCastle is shown in Figure 3. The *web crawler* collects podcasts, which can be added by users, and records them in the database manager. Those podcasts are then recognized by multiple *speech recognizers* one after another. When a request from a speech recognizer is received by the *speech recognition manager*, the next available episode to be recognized is handed over. After the recognizer finishes processing its episode, the recognition result is passed to the database manager via the speech recognition manager. The *database manager* controls the processing state of the podcasts and indexes their speech recognition results together with the corrections provided by users. Finally, the *search engine* works as a website that provides the PodCastle *user interface* with the three functions.

The web server of PodCastle was implemented by using a web application framework *Ruby on Rails 2.3.4*, a programming language *Ruby 1.8.7*, a web server *Passenger 2.2.11* and *Apache 2.2.8*, a database *MySQL Enterprise 5.0.54a*, a morphological parser for the Japanese language *ChaSen 2.3.3*, and an embeddable full-text search engine *Senna 1.1.1* and *Tritonn 1.0.9*. The client interface was implemented by using a scripting language *JavaScript 1.5* and its library *MochiKit*

[3] The confusion network was originally introduced in the context of word error minimization, which minimizes the word error rate of recognition results rather than the sentence error rate (Mangu *et al.*, 2000). Our original idea is to apply such an efficient intermediate recognition result to generate competitive candidates for efficient error correction (Ogata and Goto, 2005).

1.4, multimedia frameworks *QuickTime 10* and *Flash 10*, and an ActionScript 2 compiler *MTASC 1.12*.

To recognize podcasts, audio data is first segmented into three categories — speech, music without speech, and other background sounds — by applying GMMs. Speech segments are then recognized by using our in-house LVCSR system based on an efficient N-best search algorithm (Ogata and Ariki, 2000) to generate the confusion networks. This system uses cross-word tied-state triphone HMMs trained for 39-dimensional MFCC-based features, and a 214k-word trigram language model trained by using both large standard speech corpora and daily-updated web news.

We had to overcome various difficulties to achieve our speech recognizer for podcasts. In terms of language modeling, for example, podcasts tend to include words and phrases related to recent topics, which are usually not registered in the system vocabulary. We therefore developed a method to keep a language model up-to-date by using on-line news texts (Ogata *et al.*, 2007). In addition, in terms of acoustic modeling, podcasts include various types of speech data, such as pure speech, noisy speech, narrow-band speech, and speech with music. To reduce the acoustic mismatch, we apply several improvement methods such as noise suppression at the front-end and iterative unsupervised adaptation.

The automatic performance improvements through correction by users can be achieved through various techniques, such as those for training an acoustic model by using corrected transcriptions, for making a language model adapt to different topics by using RSS metadata and corrected transcriptions, and for registering out-of-vocabulary words by using a phonetic typewriter to estimate their pronunciation. The details of our speech recognizer are described in (Ogata *et al.*, 2007; Ogata and Goto, 2009b).

3.3 Experiences with PodCastle

PodCastle was released to the public at *http://podcastle.jp* on December 1st, 2006. PodCastle has then supported video podcasts in August 2009 by transcribing speech data in video podcasts and displaying an accompanying video screen in synchronization with the original PodCastle screen. So far, 682 podcasts have been registered, consisting of 90,985 episodes in total (as of August 31, 2010). Of these, 2242 episodes have been at least partially corrected, which resulted in 481,948 corrected words (errors). Some podcasts registered in PodCastle were corrected almost everyday or every week. We found that there are users who voluntarily cooperate in the correction, as happens with other Web 2.0 services, and that podcasts recorded by famous artists and TV personalities tend to receive many corrections.

For the collaborative training of our speech recognizer, we introduced a podcast-dependent acoustic model that is trained for each podcast using its transcripts corrected by anonymous users (Ogata *et al.*, 2007; Ogata and Goto, 2009b; Ogata and Goto, 2009a). Through our experiments, we confirmed that the speech recognition performance for some podcasts that received many error corrections was actually improved by the acoustic model training (relative error reduction of 21-33%) (Ogata and Goto, 2009b) and the burden of error correction was reduced for those podcasts. Furthermore, we are currently studying and evaluating collaborative training of language models.

We have inferred some motivations for users correcting speech recognition errors on PodCastle, though we cannot directly ask since the users are anonymous. These motivations can be categorized as follows:

- **Error correction itself is enjoyable and interesting**
  Since the error correction interface is carefully designed to be useful and efficient, using it, especially for quick and accurate operations by proficient users, could be a form of fun somewhat like a video game.
- **Users want to contribute**

Some users would often correct errors not only for their own convenience, but also to altruistically contribute to better speech recognition and retrieval.

- **Users want their podcasts to be correctly searched**
  The creators of a podcast (podcasters) would correct recognition errors in their own podcast so that it can be more accurately searched.
- **Users like the content and cannot tolerate the presence of recognition errors in it**
  Some fans of famous artists or TV personalities would correct errors because they like the podcasters' voices and cannot tolerate the presence of recognition errors in their favorite content. In fact, we have observed that such podcasts generally receive more corrections than other types.

One Web 2.0 principle is to trust users and we also trust users with respect to the quality of correction: in practice, the correction results obtained so far have been of high quality. Even if some users deliberately make inappropriate corrections (vandalism), though, we can develop countermeasures to acoustically evaluate the reliability of corrections. For example, we can use the likelihood of HMMs for forced alignment with the corrections.

4 Conclusion

We have described the PodCastle web service which provides a search engine for podcasts on the basis of the wisdom of crowds. This is the first instance of *Speech Recognition Research 2.0* which we have proposed as a new approach to speech recognition research that complements existing approaches. The technical contribution of this study is to investigate how far the performance of speech recognition and full-text search can be improved by getting speech recognition errors corrected through the cooperative efforts of many end users. At the same time, it makes a social contribution in that it helps web users by providing the world's first public web service for full-text search of Japanese-language podcasts.

Another contribution of this study is that it demonstrates how speech recognition can be put to use in situations where a speech corpus is almost impossible to prepare in advance. Although speech recognition usually requires a sufficient corpus to provide useful results, such corpora tend to be costly and labor-intensive, thus limiting applications. On the other hand, this study has aimed at *collaborative training for speech recognition* where full-text transcriptions containing recognition errors are first disclosed and then corrected by anonymous users. Since there are many errors, we run the risk of attracting criticism, but we believe that sharing these results with users will promote further popularization and use of speech recognition. We hope that this study will prove the importance and potential of incorporating user contributions into speech recognition, and that various other projects that follow the Speech Recognition Research 2.0 approach will be done, thus adding a new dimension to this field of research.

Acknowledgments

We thank Youhei Sawada, Shunichi Arai (Mellowtone Inc.), Kouichirou Eto (AIST), and Ryutaro Kamitsu (Brazil Inc.) for their web service implementation. We also thank anonymous users of PodCastle for correcting speech recognition errors.

References

Goto, Masataka, Jun Ogata, and Kouichirou Eto. 2007. PodCastle: A Web 2.0 approach to speech recognition research. In *Proceedings of the 8th Annual Conference of the International Speech Communication Association (Interspeech 2007)*, pp. 2397–2400.

Lee, Akinobu, Tatsuya Kawahara, and Kiyohiro Shikano. 2001. Julius — an open source real-time large vocabulary recognition engine. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech 2001)*, pp. 1691–1694.

Lee, Lin-Shan and Berlin Chen. 2005. Spoken document understanding and organization. *IEEE Signal Processing Magazine*, 22(5), 42–60.

Mangu, Lidia, Eric Brill, and Andreas Stolcke. 2000. Finding consensus in speech recognition: Word error minimization and other applications of confusion networks. *Computer Speech and Language*, 14(4), 373–400.

Cambridge Multimedia Document Retrieval Project. http://mi.eng.cam.ac.uk/research/projects/mdr/.

CMU Informedia Digital Video Library Project. http://www.informedia.cs.cmu.edu/.

Ogata, Jun and Yasuo Ariki. 2000. An efficient lexical tree search for large vocabulary continuous speech recognition. In *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP-2000)*, pp. 967–970.

Ogata, Jun and Masataka Goto. 2005. Speech Repair: Quick error correction just by using selection operation for speech input interfaces. In *Proceedings of the 9th European Conference on Speech Communication and Technology (Eurospeech 2005)*, pp. 133–136.

Ogata, Jun and Masataka Goto. 2009a. PodCastle: A spoken document retrieval system for podcasts and its performance improvement by anonymous user contributions. In *Proceedings of the Third Workshop on Searching Spontaneous Conversational Speech (SSCS 2009)*, pp. 37–38.

Ogata, Jun and Masataka Goto. 2009b. PodCastle: Collaborative training of acoustic models on the basis of wisdom of crowds for podcast transcription. In *Proceedings of the 10th Annual Conference of the International Speech Communication Association (Interspeech 2009)*, pp. 1491–1494.

Ogata, Jun, Masataka Goto, and Kouichirou Eto. 2007. Automatic transcription for a Web 2.0 service to search podcasts. In *Proceedings of the 8th Annual Conference of the International Speech Communication Association (Interspeech 2007)*, pp. 2617–2620.

O'Reilly, Tim. What is Web 2.0 — design patterns and business models for the next generation of software. http:// www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is- web-20.html.

Podscope. http://www.podscope.com/.

PodZinger. http://www.podzinger.com/.

Thong, Jean-Manuel Van, Pedro J. Moreno, Beth Logan, Blair Fidler, Katrina Maffey, and Matthew Moores. 2002. Speechbot: An experimental speech-based search engine for multimedia content on the web. *IEEE Transactions on Multimedia*, 4(1), 88–96.

Whittaker, Steve, Julia Hirschberg, John Choi, Don Hindle, Fernando Pereira, and Amit Singhal. 1999. SCAN: Designing and evaluating user interfaces to support retrieval from speech archives. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 26–33.

Wikipedia. http://www.wikipedia.org/.