PodCastle and Songle: Crowdsourcing-Based Web Services for Retrieval and Browsing of Speech and Music Content

Masataka Goto Jun Ogata Kazuyoshi Yoshii
Hiromasa Fujihara Matthias Mauch Tomoyasu Nakano

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

This paper describes two web services, *PodCastle* and *Songle*, that collect voluntary contributions by anonymous users in order to improve the experiences of users listening to speech and music content available on the web. These services use automatic speech-recognition and music-understanding technologies to provide content analysis results, such as full-text speech transcriptions and music scene descriptions, that let users enjoy content-based multimedia retrieval and active browsing of speech and music signals without relying on metadata. When automatic content analysis is used, however, errors are inevitable. PodCastle and Songle therefore provide an efficient error correction interface that let users easily correct errors by selecting from a list of candidate alternatives.

Keywords

Multimedia retrieval, web services, spoken document retrieval, active music listening, wisdom of crowds, crowdsourcing

1. INTRODUCTION

Our goal is to provide end users with public web services based on speech recognition, music understanding, signal processing, machine learning, and crowdsourcing so that they can experience the benefits of state-of-the-art research-level technologies. Since the amount of speech and music data available on the web is always increasing, there are growing needs for the retrieval of this data. Unlike text data, however, the speech and music data itself cannot be used as an index for information retrieval. Although metadata or social tags are often put on speech and music, annotations such as categories or topics tend to be broad and insufficient for useful content-based information retrieval [1]. Furthermore, even if users can find their favorite content, listening to it takes time. Content-based active browsing that allows random access to a desired part of the content and facilitates deeper understanding of the content is important for improving the experiences of users listening to speech and music. We therefore developed two web services for content-based retrieval and browsing: *PodCastle* for speech data and *Songle* for music data.

PodCastle (http://en.podcastle.jp for the English version and http://podcastle.jp for the Japanese version) [6, 7, 15, 16] is a spoken document retrieval service that uses automatic speech recognition (ASR) technologies to provide full-text searching of the speech data in podcasts, individual audio or movie files on the web, and the video clips on the video sharing services *YouTube*, *Nico Nico Douga*, and *Ustream.tv*). PodCastle enables users to find English and Japanese speech data including a search term, read full texts of their recognition results, and easily correct recognition errors by simply selecting from a list of candidate alternatives displayed on an error correction interface (Figure 1). The resulting corrections are used to improve the speech retrieval and recognition performance, and users can actively browse speech data by jumping to any word in the recognition results during playback. In our experience with its use over the past five years (since December 2006), over five hundred eighty thousand recognition errors were corrected by anonymous users and we confirmed that PodCastle's speech recognition performance was improved by those corrections.

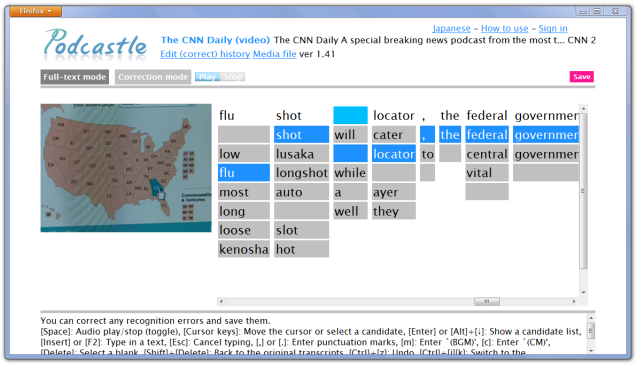


Figure 1: Screen snapshot of PodCastle's interface for correcting speech recognition errors. Competitive candidate alternatives are presented under the recognition results. A user corrected three errors in this excerpt by selecting from the candidates.

Following the success of PodCastle, we launched Songle (http://songle.jp) [8], an active music listening service that enriches music listening experiences by using music-understanding technologies based on signal processing. Songle serves as a showcase, demonstrating how people can benefit from music-understanding technologies, by enabling people to experience active music listening interfaces [5] on the web. Songle facilitates deeper understanding of music by visualizing automatically estimated music scene descriptions such as music structure, hierarchical beat structure,

Figure 2: Screen snapshot of Songle's main interface for music playback with the visualization of automatically estimated music scene descriptions.

melody line, and chords (Figure 2). Users can actively browse music data by jumping to a chorus or repeated section during playback and can use a content-based retrieval function to find music with similar vocal timbres. Songle also features an efficient error correction interface that encourages people to help improve Songle by correcting estimation errors.

2. PODCASTLE: A SPOKEN DOCUMENT RETRIEVAL SERVICE IMPROVED BY USER CONTRIBUTIONS

In 2006 we launched an ASR-based speech retrieval service, called *PodCastle* [6, 7, 15, 16], that provides full-text searching of speech data available on the web, and since then we have been improving its functions. Like the growing need for full-text search services accessing text web pages, there is a growing need for full-text speech retrieval services. Although there were previous research projects for speech retrieval [9,12,13,20,21,24] before 2006, most did not provide public web services for podcasts. There were two major exceptions, Podscope [17] and PodZinger [18], which in 2005 started web services for speech retrieval targeting English-language podcasts. They only displayed parts of speech recognition results, however, making it impossible to visually ascertain the detailed content of the speech data. And users who found speech recognition errors were offered no way to correct them. ASR technologies cannot avoid making recognition errors when processing the vast amount of speech data available on the web because speech corpora covering the diversity of topics, vocabularies, and speaking styles cannot be prepared in advance. As a result, the users of a web service using those technologies might be disappointed by its performance.

Our PodCastle web service therefore enables anonymous users to contribute by correcting speech-recognition errors. Since it provides the full text of speech recognition results, users can read those texts with a cursor moving in synchronization with the audio playback on a web browser. A user who finds a recognition error while listening can easily correct it by simply selecting an alternative from a list of candidates or typing the correct text on the error correction interface shown in Figure 1 [14]. The resulting corrections can then not only be immediately shared with other users and used to improve the spoken document retrieval performance for the corrected speech data, but also used to gradually improve the speech recognition performance by training our speech recognizer so that other speech data can be searched more reliably. This approach can be described as *collaborative training for speech-recognition technologies*.

2.1 Three Functions of PodCastle

PodCastle supports three functions: retrieving, browsing, and annotating speech data. The retrieval and browsing functions let users understand the speech recognition performance better, and the annotation (error correction) function allows them to contribute to improved performance. This improved performance can then lead to a better user experience of retrieving and browsing speech data.

2.1.1 Retrieval Function

This function allows a full-text search of speech recognition results. When the user types in a search term, a list of speech data 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. The user can access the full text of one of the search results by selecting that result and then switching over to the browsing function.

2.1.2 Browsing (Reading) Function

With this function the user can view the transcribed text of the speech data. 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 corresponding full-text result of speech recognition is available to external full-text search engines, it can be found by those engines.

2.1.3 Annotation (Error Correction) Function

This function lets users add annotations to correct any recognition errors. Here, annotation means transcribing the content of speech data, either by selecting the correct alternative from a list of competitive candidates or by typing in the correct text. On an error correction interface we earlier proposed [14] (Figure 1), 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 candidate words generated beforehand by using a *confusion network* [11] that can condense a huge internal word graph of a large vocabulary continuous speech recognition (LVCSR) system. Users do not have to worry about temporal errors in word boundaries when typing in the correct text because the temporal position of each word boundary is automatically adjusted in training the speech recognizer. Note that users are not expected to correct all the errors but to correct some errors according to their interests.

2.2 Experiences with PodCastle

The Japanese version of PodCastle was released to the public at http://podcastle.jp on December 1st, 2006 and the English version was released at http://en.podcastle.jp on October 12th, 2011. Although in the Japanese version we used AIST's speech recognizer, we have collaborated with the University of Edinburgh's Centre

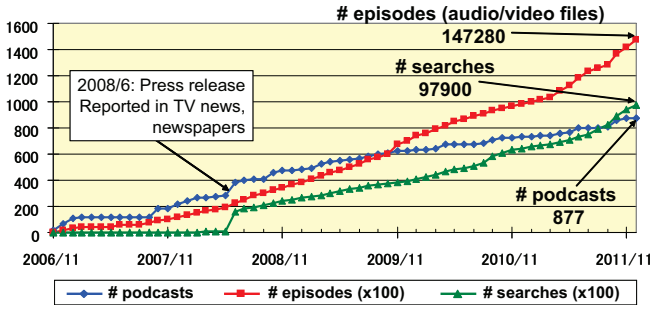


Figure 3: Cumulative usage statistics for PodCastle: the number of podcasts, the number of episodes (audio or video files), and the number of searches (queries).

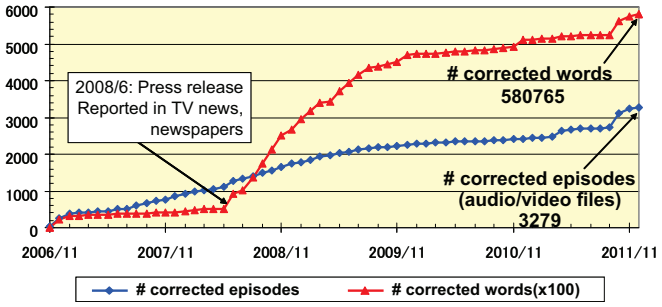


Figure 4: Cumulative usage statistics for PodCastle: the number of corrected episodes and the number of corrected words.

for Speech Technology Research (CSTR) and in the English version used their speech recognizer. In addition to supporting audio podcasts, PodCastle has supported video podcasts since 2009 and in 2011 began supporting video clips on *YouTube*, *Nico Nico Douga*, and *Ustream.tv* (recorded videos). This additional support is implemented by transcribing speech data in video clips and displaying an accompanying video screen in synchronization with the original PodCastle screen as shown in Figure 1. PodCastle has also supported functions annotating speaker names and paragraphs (new lines), marking (changing the color of) correct words that do not need any correction, and showing the percentage of correction (which becomes 100% when all the words are marked as "correct"). When several users are correcting different parts of the same speech data, those corrections can be automatically shared (synchronized) and shown on their screens. This is useful for simultaneously and rapidly transcribing speech data together.

As shown in Figure 3, 877 Japanese speech programs (such as podcasts and YouTube channels), comprising 147,280 audio files, had been registered by January 1st, 2012. Of those audio files, 3,279 had been at least partially corrected, resulting in the correction of 580,765 words (Figure 4). We found that some speech programs registered in PodCastle were corrected almost every day or every week, and we confirmed the performance was improved by the wisdom of the crowd.

For the collaborative training of our speech recognizer, we introduced a podcast-dependent acoustic model that is trained for each podcast by using transcripts corrected by anonymous users [15, 16]. Our experiments confirmed that the speech recognition performance for some podcasts that received many error corrections was improved by the acoustic model training (relative error reduction of 21-33%) [15] and that the burden of error correction was reduced for those podcasts. We also confirmed that the performance was improved by the language model training, and this will be reported in another paper.

We have inferred some motivations for users correcting errors, 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 would, especially for proficient users who master quick and accurate operations, be fun somewhat like the fun some people find in video games.
- *Users want to contribute*
  Some users would often correct errors not only for their own convenience, but also to altruistically contribute to the improvement of speech recognition and retrieval.
- *Users want their speech data to be correctly searched*
  The creators of speech data (like podcasters for podcasts) would correct recognition errors in their own speech data so that it can be searched more accurately.
- *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 speakers' voices and cannot tolerate the presence of recognition errors in their favorite content. We have indeed observed that such kinds of speech data generally receive more corrections than other kinds.

3. SONGLE: AN ACTIVE MUSIC LISTENING SERVICE IMPROVED BY USER CONTRIBUTIONS

In 2011 we launched a web service, called *Songle* [8], that allows web users to enjoy music by using *active music listening interfaces* [5], where active music listening is a way of listening to music through active interactions. In this context the word *active* does not mean that the listeners create new music but means that they take control of their own listening experience. For example, an active music listening interface called *SmartMusicKIOSK* [4] has a chorus-search function that enables a user to directly access his or her favorite part of a song (and to skip other parts) while viewing a visual representation of its music structure. This facilitates deeper understanding, but up to now the general public has not had the chance to use such research-level interfaces and technologies in their daily lives.

Toward the goal of enriching music listening experiences, Songle uses automatic music-understanding technologies to estimate music scene descriptions (musical elements) [3] of musical pieces (audio files) available on the web. A Songle user can enjoy playing back a musical piece while seeing the visualization of the estimated descriptions. In our current implementation, four major types of descriptions are automatically estimated and visualized for content-based music browsing: music structure (chorus sections and repeated sections), hierarchical beat structure (musical beats and bar lines), melody line (fundamental frequency (F0) of the vocal melody), and chords (root note and chord type). Songle implements all functions that the interface of SmartMusicKIOSK had and lets a user jump and listen to the chorus by just pushing the next-chorus button. Songle thus makes it easier for a user to find desired parts of a piece.

Given the variety of musical pieces on the web, however, music scene descriptions are hard to estimate accurately. Because of the diversity of music genres and recording conditions and the complexity of sound mixtures, automatic music-understanding tech-

nologies cannot avoid making some errors. As a result, the users of a web service using those technologies might be disappointed by its performance.

Our Songle web service therefore enables anonymous users to help improve its performance by correcting music-understanding errors. Each user can see the music-understanding visualizations on a web browser, where a moving cursor indicates the audio playback position. A user who finds an error while listening can easily correct it by selecting from a list of candidate alternatives, or by providing an alternative description via an error correction interface. The resulting corrections are then shared and used to immediately improve user experience with the corrected piece. We also plan to use such corrections to gradually improve music-understanding technologies through adaptive machine learning techniques so that descriptions of other musical pieces can be estimated more accurately. This approach can be described as *collaborative training for music-understanding technologies*.

The alpha version of Songle was released to the public at http://songle.jp on October 20th, 2011. During the initial stage of the Songle launch we are focusing on popular songs with vocals. A user can register any song available on the web by providing the URL of its MP3 file, the URL of a web page including multiple MP3 URLs, or the URL of a music podcast (an RSS syndication feed including multiple MP3 URLs). In addition to contributing to the enrichment of music listening experiences, Songle will serve as a showcase in which everybody can experience music-understanding technologies and understand their nature: for example, what kinds of music or sound mixture are difficult for the technologies to handle.

3.1 Three Functions of Songle

Songle supports three main functions: retrieving, browsing, and annotating songs. The retrieval and browsing functions facilitate deeper understanding of music, and the annotation (error correction) function allows users to contribute to the improvement of music scene descriptions. The improved descriptions can lead to a better user experience of retrieving and browsing songs.

3.1.1 Retrieval Function

This function enables a user to retrieve a song by making a text search for the song title or artist name or by making a selection from a list of artists or a list of songs whose descriptions were recently estimated or corrected. This function also shows various kinds of rankings.

Following the idea of an active music listening interface *VocalFinder* called [2], which finds songs with similar vocal timbres, Songle provides a similarity graph of songs so that a user can retrieve a song according to vocal timbre similarity. The graph is a radially connected network in which nodes (songs) of similar vocal timbre are connected to the center node (a recommended or user-specified song). By traversing a graph while listening to nodes, a user can find a song having the favorite vocal timbre.

By selecting a song, the user switches over to the within-song browsing function.

3.1.2 Within-song Browsing Function

This function provides a content-based playback-control interface for within-song browsing as shown in the upper half of Figure 2. The upper window is the global view showing the entire song and the lower window is the local view magnifying the selected region. A user can view the following four types of music scene descriptions estimated automatically:

1. *Music structure (chorus sections and repeated sections)*
   In the global view, the *music map* of the SmartMusicKIOSK interface [4] is shown below the playback controls including the buttons, time display, and playback slider. The music map is a graphical representation of the entire song structure and consists of chorus sections (the top row) and repeated sections (the five lower rows). On each row, colored sections indicate similar (repeated) sections. Clicking directly on a colored section plays that section.
2. *Hierarchical beat structure (musical beats and bar lines)*
   At the bottom of the local view, musical beats corresponding to quarter notes are visualized by using small triangles. Bar lines are marked by larger triangles.
3. *Melody line (F0 of the vocal melody)*
   The piano roll representation of the melody line is shown above the beat structure in the local view. It is also shown in the lower half of the global view. For simplicity, the fundamental frequency (F0) can be visualized after being quantized to the closest semitone.
4. *Chords (root note and chord type)*
   Chord names are written in the text at the top of the local view. Twelve different colors are used to represent twelve different root notes so that a user can notice the repetition of chord progressions.

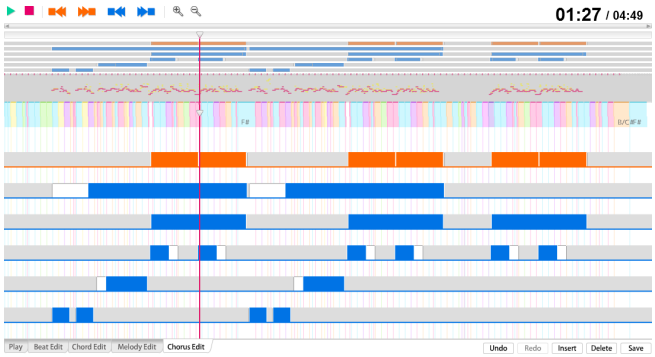
3.1.3 Annotation (Error Correction) Function

This function allows users to add annotations to correct any estimation errors. Here, annotation means describing the contents of a song, either by modifying the estimated descriptions or by selecting the correct candidate if it is available. In the local view, a user can switch between editors for four types of music scene descriptions.

1. *Music structure* (Figure 5(a))
   The beginning and end points of every chorus or repeated section can be adjusted. It is also possible to add, move, or delete each section. This correction function improves the SmartMusicKIOSK experience.
2. *Hierarchical beat structure* (Figure 5(b))
   Several alternative candidates for the beat structure can be selected at the bottom of the local view. If none of the candidates are appropriate, a user can enter the beat position by tapping a key during music playback. Each beat position or bar line can also be changed directly. For fine adjustment it is possible to play the audio back with click tones at beats.
3. *Melody line* (Figure 5(c))
   Songle allows note-level correction on the piano roll representation of the melody line. Since the melody line is internally represented as the temporal trajectory of F0, more precise correction is also possible. More accurate melody annotations will lead to better similarity graphs of songs.
4. *Chords* (Figure 5(d))
   Chord names can be corrected by choosing from candidates or by typing in chord names. Each chord boundary can also be adjusted. Chords can be played back along with the original song to make it easier to check the correctness.

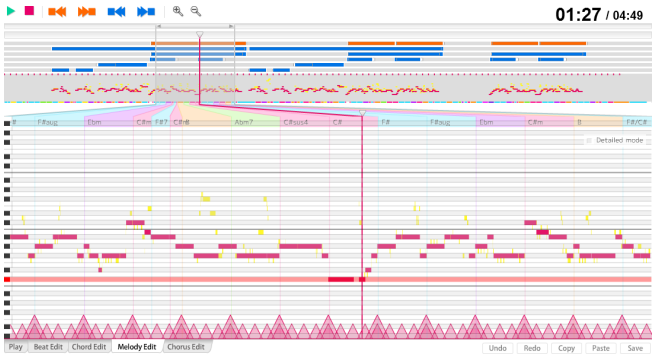
Note that users can simply enjoy active music listening without correcting errors. We understand that it is too difficult for some users to correct the above descriptions (especially, chords). Designing an interface that makes it easier for them to make corrections will be another future challenge. Moreover, users are not expected to correct all errors, only some according to each user's interests.

When the music-understanding results are corrected by users, the original values are visualized as trails with different colors (white,

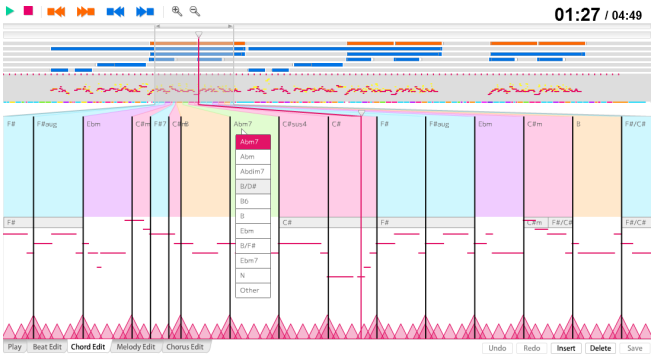


(a) Correcting music structure
(chorus sections and repeated sections)

(b) Correcting hierarchical beat structure
(musical beats and bar lines)



(c) Correcting melody line (F0 of the vocal melody)



(d) Correcting chords (root note and chord type)

Figure 5: Screen snapshots of Songle's annotation function for correcting music scene descriptions.

gray, or yellow marks in Figure 5) that can be distinguished by anybody. These trails are important to prevent overestimation of the automatic music-understanding performance after the user corrections. Moreover, all the correction histories are recorded, and descriptions before and after corrections can be compared.

4. DISCUSSION

We discuss how PodCastle and Songle could contribute to society and academic research.

4.1 Contributions of PodCastle and Songle

PodCastle and Songle make social contributions by providing public web services that let people retrieve speech data by using speech-recognition technologies and that let people enjoy active music listening interfaces with music-understanding technologies. They also promote the popularization and use of speech-recognition and music-understanding technologies by raising user awareness. Users can grasp the nature of those technologies just by seeing results obtained when the technologies applied to speech data and songs available on the web. We risk attracting criticism when there are many errors, but we believe that sharing these results with users will promote the popularization of this research field.

PodCastle and Songle make academic contributions by demonstrating a new research approach to speech recognition and music understanding based on signal processing; this approach aims to improve the speech-recognition and music-understanding performances as well as the usage rates while benefiting from the cooperation of anonymous end users. This approach is designed to set into motion a *positive spiral* where (1) we enable users to experience a service based on speech recognition or music understanding to let them better understand its performance, (2) users contribute to improving performance, and (3) the improved performance leads to a better user experience, which encourages further use of the service at step (1) of this spiral. This is a *social correction* framework, where users can improve the performance by sharing their correction results over a web service. The game-based approach of Human Computation or GWAPs (games with a purpose) [22] like the ESP Game [23] often lacks step (3) and depends on the feeling of fun. In this framework, users gain a real sense of contributing for their own benefit and that of others and can be further motivated to contribute by seeing corrections made by other users. In this way, we can use the *wisdom of the crowd* or *crowdsourcing* to achieve a better user experience.

Another important technical contribution is that PodCastle and Songle let us investigate how much the performance of speech-recognition and music-understanding technologies can be improved by getting errors corrected through the cooperative efforts of users. Although we have already implemented a machine-learning mechanism to improve the performance of the speech-recognition technology on the basis of user corrections on PodCastle, we have not yet implemented such a mechanism to improve the performance of the music-understanding technology on the basis of user corrections on Songle because it has just recently been launched. When we have collected enough corrections, we could also implement such a mechanism on Songle. This study thus provides a framework for *amplifying* user contributions. In a typical *Web 2.0* service like *Wikipedia*, improvements are limited to an item directly contributed (edited) by users. In PodCastle, the improve-

ment of the speech recognition performance automatically spreads improvements to items not contributed by users. In Songle, improvements will also spread to other songs when we will implement the improvement mechanism. This is a novel technology of amplifying user contributions, which could be beyond Web 2.0 and Human Computation [22]. We hope that this study will show the importance and potential of incorporating and amplifying user contributions and that various other projects [10, 19] that follow this approach will be done, thus adding a new dimension to this field of research.

One Web 2.0 principle is to trust users, and we think users can also be trusted with respect to the quality of their corrections. In fact, as far as we assessed the quality, the correction results obtained so far have been of high quality. One of the reasons would be that PodCastle and Songle avoid relying on monetary rewards as Amazon Mechanical Turk does. Even if some users make inappropriate corrections deliberately (the vandalism problem), we will be able to develop countermeasures evaluating the reliability of corrections acoustically. For example, we could validate whether the corrected descriptions can be supported by acoustic phenomena. This will be another interesting research topic.

4.2 PodCastle and Songle as a Research Platform

We hope to extend PodCastle and Songle to serve as a research platform where other researchers can also exhibit the results of their own speech-recognition and music-understanding technologies. Since even in our current implementations of PodCastle and Songle a module of each technology can be executed anywhere in the world, its source and binary codes need not be shared. Its module can just connect to our web server to receive an audio file and send back speech-recognition or music-understanding results via HTTP. The results should always be shown with clear acknowledgments/credits so that users can distinguish the sources.

This platform is especially useful for supporting various languages for PodCastle. In fact, the English version of PodCastle was implemented in this platform and the CSTR's speech recognizer for English language is executed at CSTR, University of Edinburgh.

5. CONCLUSION

We have described PodCastle, a spoken document retrieval service that provides a search engine for web speech data and is based on the wisdom of the crowd (crowdsourcing), and Songle, an active music listening service that is continually improved by anonymous user contributions. In our current implementations, full-text transcriptions of speech data and four types of music scene descriptions are recognized, estimated, and displayed through web-based interactive user interfaces. Since automatic speech-recognition and music-understanding technologies are not perfect, PodCastle and Songle allow users to make error corrections that are shared with other users, thus creating a positive spiral and giving users an incentive to keep making corrections. This platform will act both as a test-bed or showcase for new technologies and as a way of collecting valuable annotations.

Acknowledgments: We thank Youhei Sawada, Shunichi Arai, Kouichirou Eto, and Ryutaro Kamitsu for their web service implementation of PodCastle, Utah Kawasaki for the web service implementation of Songle, and Minoru Sakurai for the web design of PodCastle and Songle. We also thank anonymous users of PodCastle and Songle for correcting errors. This work was supported in part by CREST, JST.

6. REFERENCES

[1] M. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, 96(4):668–696, 2008.

[2] H. Fujihara, M. Goto, T. Kitahara, and H. G. Okuno. A modeling of singing voice robust to accompaniment sounds and its application to singer identification and vocal-timbre-similarity-based music information retrieval. *IEEE Trans. on ASLP*, 18(3):638–648, 2010.

[3] M. Goto. A real-time music scene description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals. *Speech Communication*, 43(4):311–329, 2004.

[4] M. Goto. A chorus-section detection method for musical audio signals and its application to a music listening station. *IEEE Trans. on ASLP*, 14(5):1783–1794, 2006.

[5] M. Goto. Active music listening interfaces based on signal processing. In *Proc. of ICASSP 2007*, 2007.

[6] M. Goto and J. Ogata. PodCastle: Recent advances of a spoken document retrieval service improved by anonymous user contributions. In *Proc. of Interspeech 2011*, 2011.

[7] M. Goto, J. Ogata, and K. Eto. PodCastle: A Web 2.0 approach to speech recognition research. In *Proc. of Interspeech 2007*, 2007.

[8] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano. Songle: A web service for active music listening improved by user contributions. In *Proc. of ISMIR 2011*, pages 311–316, 2011.

[9] L. Lee and B. Chen. Spoken document understanding and organization. *IEEE Signal Processing Magazine*, 22(5):42–60, 2005.

[10] S. Luz, M. Masoodian, and B. Rogers. Supporting collaborative transcription of recorded speech with a 3D game interface. In *Proc. of KES 2010*, 2010.

[11] L. Mangu, E. Brill, and A. Stolcke. Finding consensus in speech recognition: Word error minimization and other applications of confusion networks. *Computer Speech and Language*, 14(4):373–400, 2000.

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

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

[14] J. Ogata and M. Goto. Speech Repair: Quick error correction just by using selection operation for speech input interfaces. In *Proc. of Eurospeech 2005*, pages 133–136, 2005.

[15] J. Ogata and M. Goto. PodCastle: Collaborative training of acoustic models on the basis of wisdom of crowds for podcast transcription. In *Proc. of Interspeech 2009*, pages 1491–1494, 2009.

[16] J. Ogata, M. Goto, and K. Eto. Automatic transcription for a Web 2.0 service to search podcasts. In *Proc. of Interspeech 2007*, 2007.

[17] Podscope. http://www.podscope.com/.

[18] PodZinger. http://www.podzinger.com/.

[19] N. Ramzan, M. Larson, F. Dufaux, and K. Cluver. The participation payoff: Challenges and opportunities for multimedia access in networked communities. In *Proc. of ACM MIR 2010*, 2010.

[20] J.-M. V. Thong, P. J. Moreno, B. Logan, B. Fidler, K. Maffey, and M. Moores. Speechbot: An experimental speech-based search engine for multimedia content on the web. *IEEE Trans. on Multimedia*, 4(1):88–96, 2002.

[21] V. Turunen, M. Kurimo, and I. Ekman. Speech transcription and spoken document retrieval in Finnish. *Machine Learning for Multimodal Interaction*, 3361:253–262, 2005.

[22] L. von Ahn. Games with a purpose. *IEEE Computer Magazine*, 39(6):92–94, June 2006.

[23] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proc. of CHI 2004*, pages 319–326, 2004.

[24] S. Whittaker, J. Hirschberg, J. Choi, D. Hindle, F. Pereira, and A. Singhal. SCAN: Designing and evaluating user interfaces to support retrieval from speech archives. In *Proc. of ACM SIGIR 99*, pages 26–33, 1999.