PodCastle: Collaborative Training of Language Models on the Basis of Wisdom of Crowds

Jun Ogata and Masataka Goto

National Institute of Advanced Industrial Science and Technology (AIST)
1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, JAPAN

Abstract

This paper presents a language-model training method for improving automatic transcription of online spoken contents. Unlike previously studied LVCSR tasks such as broadcast news and lectures, large-sized task-specific corpora for training language models cannot be prepared and used in recognition because of the diversity of topics, vocabularies, and speaking styles. To overcome difficulties in preparing such task-specific language models in advance, we propose collaborative training of language models on the basis of wisdom of crowds. On our public web service for LVCSR-based spoken document retrieval *PodCastle*, over half a million recognition errors were corrected by anonymous users. By leveraging such corrected transcriptions, component language models for various topics can be built and dynamically mixed to generate an appropriate language model for each podcast episode in an unsupervised manner. Experimental results with Japanese podcasts showed that the mixed languages models significantly reduced the word error rate.

Index Terms: web service, LVCSR, language modeling, wisdom of crowds, error correction

1. Introduction

Given ever-increasing speech data included in podcasts and video clips on the web, there is a great demand for web services for spoken document retrieval. We therefore developed and launched a public web service, *PodCastle* [1][2][3][4][5], that is based on automatic speech recognition (ASR) technologies. PodCastle (http://en.podcastle.jp for the English version and http://podcastle.jp for the Japanese version) provides full-text searching of the speech data in podcasts, individual audio or movie files on the web, and video clips on video sharing services. PodCastle enables users to find speech data including a search term, read full texts of their recognition results (transcripts) with a cursor moving in synchronization with the audio playback on a web browser, 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.

To improve the usefulness of this web service for end users, we have studied and developed several speech recognition methods to increase the accuracy of transcripts of the online spoken content [2]. Recent advances in state-of-the-art large vocabulary continuous speech recognition (LVCSR) systems can be attributed to the availability of large amounts of training data. In usual, these training data are collected for each specific task like broadcasting news, lectures, meetings, etc., and their accurate transcriptions are manually prepared. The most critical issue in transcribing online spoken content such as podcasts and video clips is that there is a wide variety of both linguistic content and acoustic conditions. A common approach of building a task-specific corpus in advance is impractical because it will be too costly and time consuming to prepare a corpus that covers the diversity of the content and acoustic conditions. To overcome this difficulty with the focus on acoustic modeling, we have proposed an acoustic-model training method for podcast transcription in our previous study [3], but have not described a language-model training method that leverages user corrections.

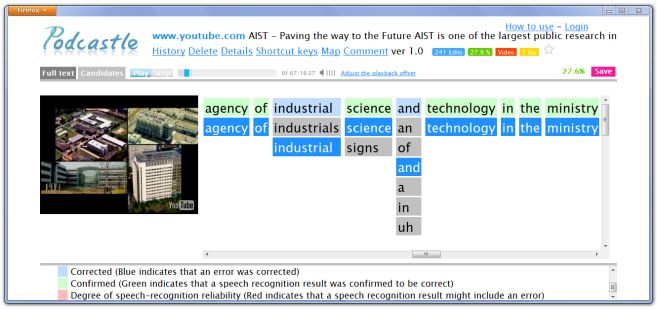

Figure 1: Screen snapshot of PodCastle's interface for correcting speech recognition errors. Competitive candidate alternatives are presented under the 1-best recognition results.

This paper focuses on language modeling to improve recognition performance because language modeling is another important, challenging issue in transcribing podcasts and other similar content such as YouTube video clips featuring spoken documents. In the literature, various works have been done in terms of language model (LM) adaptation for several LVCSR tasks like broadcast news [6][7], meetings [8][9], and lectures [10][11]. These works basically took an approach that a main (or background) LM trained from large amounts of task-specific text data is adapted using an additional resource such as in-domain (on-topic) text data [7][8], web-based text data [6][11], or user-provided text data [9][10]. In our target having the diversity of topics, however, such a large-scale task-specific corpus cannot be prepared in advance to train the background LM.

We therefore study the use of the benefits of our web service PodCastle for generating the content-specific LM training data. PodCastle encourages users to cooperate by correcting speech-recognition errors so that speech data can be searched more reliably. If a lot of users contribute to the error correction, PodCastle can provide relatively accurate transcripts of spoken content. We incorporate such *wisdom of crowds* from the users' contributions into language-model training. In other words, the transcripts voluntarily corrected by anonymous users are used for language-model training. Furthermore, we explore the use of a large number of text articles from a news aggregation website that covers various topics and vocabularies for the background LM. Our language modeling approach is based on a dynamic mixture of multiple component LMs that are trained from two kinds of resources: topically classified news articles and the user transcripts on PodCastle.

Table 1: Number of words corrected by anonymous users and the breakdown by error correction procedure on the Japanese version of PodCastle (as of January 1st, 2012).

Number of corrected words	580,765
Averaged number of corrected words per episode	177
Correction procedure of selecting from candidates	300,603
Correction procedure of typing in the correct text	280,162

Table 2: Amount of web news text for each sub-topic in *Yahoo! Japan News*. There are 6 main topics: *Business* (Bus.), *Domestic* (Dom.), *International* (Int.), *Entertainment* (Ent.), *Sports* (Spo.), and *Technology* (Tec.), and each main topic consists of several sub-topics as shown in parentheses.

Topic (Sub)	#words	Topic (Sub)	#words
Bus. (Market)	8.3M	Ent. (Video game)	9.3M
Bus. (Stock)	10.4M	Spo. (Baseball)	23.4M
Bus. (Industry)	23.5M	Spo. (Soccer)	14.3M
Bus. (Other)	55.7M	Spo. (Motor spo.)	5.6M
Dom. (Politics)	19.3M	Spo. (Horse race)	5.9M
Dom. (Society)	65,3M	Spo. (Golf)	7.4M
Dom. (People)	0.7M	Spo. (Fight spo.)	8.8M
Int. (China)	16.6M	Spo. (Other)	50.1M
Int. (Korea)	9.0M	Tec. (Internet)	7.3M
Int. (Other)	32.7M	Tec. (Mobile device)	5.9M
Ent. (Music)	14.0M	Tec. (Security)	2.2M
Ent. (Movie)	10.6M	Tec. (Science)	48.7M
Ent. (Other)	43.1M		

2. Usage Analysis of 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 our speech recognition system, we have collaborated with the University of Edinburgh's Centre for Speech Technology Research (CSTR) and in the English version used their speech recognition system. 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.

In the Japanese version of PodCastle, 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. 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. Motivations for users correcting errors were discussed in [4][5]. Table 1 shows the breakdown according to the error correction procedure used on PodCastle. We found that the correction procedure of selecting from a list of candidate alternatives is more popular than the correction procedure of typing in the correct text. According to a detailed usage history, we also confirmed that users tend to choose the candidate-selection procedure if correct words are included in candidate alternatives.

3. Dynamic Language Modeling System

To make use of user corrections as described above, we propose a dynamic language modeling system with the aim of deploying its system on the PodCastle web service in the near future. The basic idea is to dynamically mix a *channel-dependent LM*, which is trained from transcriptions (ASR results) that were voluntarily corrected by anonymous users on PodCastle, with background LMs for each target podcast episode (speech data). As linguistic resources for the background LMs, we focus on text data of web news articles that cover various topics and vocabularies. Differences of topics in web news articles are appropriately considered in our dynamic language modeling.

3.1. Web-news-based topic LMs

In generating background LMs for online spoken content, a large number of text articles from a news aggregation website are useful. Such text data, which we refer to *web news text*, offers two main benefits in language modeling. First, there are various kinds of news articles that come from different news sites, and all the articles are structured and classified into many categories to make crawling easier. Second, hot topics with the latest words and phrases can be covered since such articles are frequently updated on a daily basis.

In this work, we use topically classified text data obtained from *Yahoo! Japan News* (http://headlines.yahoo.co.jp). In Yahoo! Japan News, all of the articles are classified into hierarchical topic trees that consist of 6 main topics and 25 sub-topics. In our current implementation, we train a language model for each sub-topic by using news articles published during a 40 month period (from February 2007 to June 2010). Table 2 shows the amount of text data for each sub-topic. In the following, we refer to these sub-topic level LMs as *topic LMs*.

3.2. Channel-dependent LM

Speech recognition results (transcripts) of similar spoken content are also useful for language modeling. In our previous work [3], we developed a podcast-dependent (channel-dependent) acoustic modeling method to deal with various acoustic conditions in different podcasts under the assumption that episodes belonging to each podcast have a similar acoustic condition. In the same way, we assume that episodes (speech data) belonging to each channel (such as podcast, YouTube channel, and Ustream channel) have a linguistic similarity in terms of topics, vocabularies, and frequent phrases. Transcripts of episodes within the same channel can therefore be useful as an additional text resource for speech recognition.

As a component LM used when recognizing each episode of a channel, a channel-dependent LM specialized for its episode is generated on the fly by using all the other episodes within the same channel. If speech-recognition errors in transcripts of those episodes are corrected by users on PodCastle, the resulting channel-dependent LM is expected to be improved.

3.3. Dynamic mixture of topic LMs

The overall process of our dynamic language modeling system is shown in Figure 2. The system is based on a model-level mixture scheme that combines different web-news-based topic LMs with a channel-dependent LM generated by using transcripts of

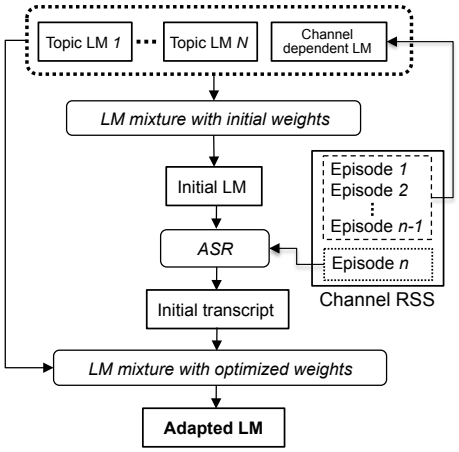

Figure 2: Overview of the proposed dynamic language modeling (LM adaptation) system that will be incorporated into the PodCastle web service in the future. Each web-news-based topic LM is interpolated with spontaneous text data (CSJ) to deal with a spontaneous speaking style.

other episodes. In the model-level mixture, the n-gram probabilities of multiple component LMs are combined as follows:

$$p_{mix}(w|h) = \sum_i \lambda_i p_i(w|h) \quad (1)$$

where λ_i are non-negative mixture weights such that $\sum_i \lambda_i = 1$. The mixture weights for component LMs are computed by minimizing the perplexity on development data via the Expectation-Maximization (EM) algorithm.

In the static phase, we first train 26 component LMs that consist of 25 topic LMs and a channel-dependent LM. To reflect characteristics of spontaneous speech, another component LM trained with lecture transcripts of the Corpus of Spontaneous Japanese (CSJ) [12] is interpolated with each web-news-based topic LM. In our current implementation, the interpolation weight is fixed at 0.5. We then generate a multi-topic initial LM by mixing the 26 component LMs with the equivalent (uniform) mixture weights. The vocabulary of the initial LM should be relatively large (over 280K words in our system) since online spoken content has various topics in wide domains.

In the dynamic phase, we then generate the initial transcript for the input episode by using our ASR system with the above initial LM. The mixture weights for the component LMs are then re-computed by minimizing the perplexity on the initial transcript as the development data. Finally the LM topically adapted to the episode is generated from a mixture of the component LMs according to the re-computed weights.

4. Experiments

To investigate the effectiveness of our system, we conducted recognition experiments using actual podcast speech data.

4.1. Experimental conditions

To make it easier to incorporate our system into the PodCastle web service, we tested it on excerpts of transcripts corrected by users on PodCastle.

4.1.1. Speech data

For evaluation, we used 45 episodes in total from 8 Japanese podcasts as shown in Table 3. These podcasts were actually registered by users on the PodCastle web service. The test set included three domains: daily news distributed by a Japanese broadcasting company (A, B), lecture-style talks (C, D, E, F), and chitchat shows (G, H). The main topic varied with each podcast as shown in the table. The news podcasts (A, B) included several kinds of topics within one episode. Some episodes of each podcast were voluntarily corrected by users on PodCastle as mentioned in Section 2. As the training data for generating the channel-dependent LMs (Table 3), we used all the episodes that were at least partially corrected on PodCastle.

4.1.2. Decoding system

To deal with a continuous audio stream of each episode, audio data is first segmented into three categories (speech, music without speech, and other background sounds) by applying GMMs [2]. Speech segments are then decoded in our two pass recognition system [2] implemented for PodCastle, and finally a confusion network is generated. As a recognition hypothesis, the 1-best word sequence of the confusion network is extracted.

As an acoustic model, a tied-state triphone HMM was trained with 600 h of presentation speech data from the CSJ corpus [12]. The acoustic features are 39 PLP-cepstral coefficients, and CMLLR-based speaker adaptive training (SAT) [13] was done.

4.2. Results

Table 4 summarizes the recognition performance of our dynamic LM adaptation system. The baseline performance in this table was obtained using the mixture LM with initial weights ($\lambda_i = 1/26$). Although all the results were obtained by the recognition system described in Section 4.1.2, we used the same acoustic model which was adapted with the recognition hypothesis of the baseline LM system ("Initial transcript" in Figure 2). From the result of "w/o podcast LM", it is shown that the dynamic mixture of web-news-based topic LMs can consistently reduce the word error rate (WER) for every podcast in the test set. The mixture weight optimization method in our system is equivalent to a selection of promising topics with a minimum perplexity criterion for each podcast episode. In the case of podcasts where there was relatively large improvement (B, G), the appropriate topics were selected; i.e., the value of λ_i was outstanding among the component LMs.

When channel-dependent LM without corrections was used, the WER was further reduced from 33.5% to 32.7% on average. This suggests that the availability of sufficient training data for each podcast is important in language modeling even though the transcriptions are not perfect. By using channel-dependent LMs with the error corrections, the recognition performance was significantly improved for all the podcasts. It is shown that the error corrections and the transcriptions obtained from PodCastle are very effective in language modeling for online spoken content.

Finally, in order to investigate the effect of the recognition errors in mixture weight optimization, we carried out a supervised experiment in which the mixture weights were optimized using the manual transcripts of the test-set episodes. Although the system with the supervised weight optimization reduced the WER, the difference in the performance from the unsupervised system was not so large (Table 4). This suggests that the mixture weight optimization of these 26 component LMs is relatively robust to recognition errors.

5. Conclusion

In this paper, we have described a language-model training method for improving transcription of online spoken content.

Table 3: Description of podcast data used in the experiment. The training data was used to generate channel-dependent LMs. The topic *"multi"* indicates that various kinds of topics were included in a podcast.

ID	Domain	Main topic	Test data #episodes (#words)	Training data #episodes (#words)
A	news	*multi*	4 (11,170)	94 (259,840)
B	news	*multi*	4 (4,937)	23 (24,908)
C	lecture	politics	20 (13,876)	79 (65,984)
D	lecture	business	5 (10,763)	35 (125,432)
E	quiz show	general knowledge	2 (1,910)	30 (39,098)
F	lecture	health & care	2 (3,292)	71 (193,569)
G	chitchat	music	4 (25,239)	56 (283,414)
H	chitchat	show business	4 (9,251)	60 (127,840)

Table 4: WER (word error rate %) results for podcast transcription task. The channel-dependent LM is referred to as *"channel LM"*.

ID	Baseline	Proposed dynamic language modeling system		
		w/o *channel LM*	w/ *channel LM*	
			w/o corrections	w/ corrections
A	17.9	16.4	15.8	**14.9**
B	21.3	19.3	17.4	**16.3**
C	28.2	27.2	26.3	**23.7**
D	41.1	39.8	38.1	**36.9**
E	35.8	35.0	33.3	**27.9**
F	29.7	28.2	25.1	**22.4**
G	31.1	29.1	28.5	**24.2**
H	59.5	58.8	58.2	**54.0**
Ave.	35.3	33.5	32.7	**28.9**

Table 5: WER (%) comparison of unsupervised (with recognition hypothesis) and supervised (with manual transcriptions) weight optimization.

	unsupervised	supervised
w/o channel LM	33.5	33.4
w/ channel LM	28.9	28.6

To overcome the difficulties dealing with a variety of topics on a typical corpus-based LVCSR system, we incorporated the wisdom of crowds, i.e., error corrections made by anonymous users, into language modeling. We developed a dynamic language modeling system in which the topic LMs based on web news and the channel-dependent LM generated by transcripts corrected by users are dynamically mixed with optimal weights. The experimental results have shown that our method can improve the recognition performance in podcast transcription.

The current PodCastle web service simply uses a LM adaptation method in which the baseline (initial) LM is interpolated with the channel-dependent LM. In the future, we plan to deploy the proposed dynamic language modeling system on the PodCastle web service to improve its ASR performance and derive the full benefit from user corrections.

6. Acknowledgements

We thank the Centre for Speech Technology Research (CSTR) at the University of Edinburgh for the English version of PodCastle. This study was supported in part by the JSPS KAKENHI 23700225.

7. References

[1] M. Goto, J. Ogata, and K. Eto, "PodCastle: A Web 2.0 approach to speech recognition research," in *Proc. of Interspeech 2007*, 2007, pp. 2397–2400.

[2] J. Ogata, M. Goto, and K. Eto, "Automatic transcription for a web 2.0 service to search podcasts," in *Proc. of Interspeech 2007*, 2007, pp. 2617–2620.

[3] 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*, 2009, pp. 1491–1494.

[4] 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.

[5] M. Goto, J. Ogata, K. Yoshii, H. Fujihara, M. Mauch and T. Nakano, "PodCastle and Songle: Crowdsourcing-Based Web Services for Retrieval and Browsing of Speech and Music Content," in *Proc. of CrowdSearch 2012*, 2012.

[6] M. Federico and N. Bertoldi, "Broadcast news lm adaptation over time," *Computer Speech & Language*, vol. 18, pp. 417–435, 2004.

[7] X. Lei, W. Wu, W. Wang, A. Mandal, and A. Stolcke, "Development of the 2008 SRI mandarin speech-to-text system for broadcast news and conversation," in *Proc. of Interspeech 2009*, 2009.

[8] G. Tur and A. Stolcke, "Unsupervised language model adaptation for meeting recognition," in *Proc. ICASSP2007*, 2007.

[9] D. Vergyri, A. Stolcke, and G. Tur, "Exploiting user feedback for language model adaptation in meeting recognition," in *Proc. of ICASSP 2009*, 2009.

[10] B.-J. P. Hsu and J. Glass, "Language model parameter estimation using user transcription," in *Proc. of ICASSP 2009*, 2009.

[11] S. Meng, K. Thambiratnam, Y. Lin, L. Wang, G. Li, and F. Seide, "Vocabulary and language model adaptation using just one speech file," in *Proc. of ICASSP 2010*, 2010.

[12] T. Kawahara, H. Nanjo, T. Shinozaki, and S. Furui, "Benchmark test for speech recognition using the corpus of spontaneous Japanese," in *Proc. SSPR 2003*, 2003.

[13] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech & Language*, vol. 12, pp. 75–98, 1998.

[14] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: Word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.