Two-stream Indexing for Spoken Web Search

Jitendra Ajmera[1], Anupam Joshi[2], Sougata Mukherjea[1], Nitendra Rajput[1], Shrey Sahay[1], Mayank Shrivastava[3], Kundan Srivastava[1]

[1]IBM Research
4, Block C, ISID Campus,
Vasant Kunj, New Delhi, India
jajmera1,smukherj,rnitendra,
kunsriva,shrsahay@in.ibm.com

[2]University of Maryland
Baltimore County
Maryland, USA
joshi
@umbc.edu

[3]IIT Kharagpur
Kharagpur
West Bengal, India
msrivastava
@cse.iitkgp.ernet.in

ABSTRACT

This paper presents two-stream processing of audio to index the audio content for Spoken Web search. The first stream indexes the meta-data associated with a particular audio document. The meta-data is usually very sparse, but accurate. This therefore results in a *high-precision, low-recall* index. The second stream uses a novel language-independent speech recognition to generate text to be indexed. Owing to the multiple languages and the noise in user generated content on the Spoken Web, the speech recognition accuracy of such systems is not high, thus they result in a *low-precision, high-recall* index. The paper attempts to use these two complementary streams to generate a combined index to increase the precision-recall performance in audio content search.
The problem of audio content search is motivated by the real world implication of the Web in developing regions, where due to literacy and affordability issues, people use Spoken Web which consists of interconnected VoiceSites, which have content in audio. The experiments are based on more than 20,000 audio documents spanning over seven live VoiceSites and four different languages. The results suggest significant improvement over a *meta-data-only* or a *speech-recognition-only* system, thus justifying the two-stream processing approach. Audio content search is a growing problem area and this paper wishes to be a first step to solving this at a large scale, across languages, in a Web context.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering; H.5.1 [**Multimedia Information Systems**]: Audio input/output; H.5.4 [**Hypertext/Hypermedia**]: Navigation

General Terms

Algorithms, Design, Experimentation, Human Factors, Languages

Keywords

World Wide Telecom Web, Spoken Web, developing regions, mobile phone, literacy, audio search

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0637-9/11/03.

1. INTRODUCTION

The growth of the Internet has been consistently followed by the growth of the search engines. The number of Internet users have increased more than ten times over the last ten years. Further, with the Web 2.0 phenomenon, the increase in the amount of content being generated on the Internet has also been significant. The size of an average web page has more than quintupled since 2003 [11]. With such increased content on the web, simply browsing it becomes challenging. It is therefore clear that search engines become critical for the proliferation of Web when there is a large amount of content being created on the Internet.

As a corollary, and focusing on developing regions, we wish to position that the search of audio content will be critical when we focus on providing Web-like features to people in the developing world. Due to low-literacy of the population [33] and increasing penetration of cell phones, audio interactions over phone are being considered as an alternate to the PC model of information access on the Internet. Over the last couple of years, there have been several projects that rely on user-generated audio-content for information dissemination tasks [29, 30, 34, 25]. These systems establish the usefulness of speech interfaces for the developing world.

The World Wide Telecom Web (or Spoken Web) [22, 20, 29] is a system that leverages the pervasiveness of mobile phones in developing countries and allows the creation, deployment and hosting of voice-driven applications called VoiceSites by any phone subscriber. It attempts to empower users by giving them the opportunity to become content creators and provides a mechanism for users to access information and content with affordable devices (just a regular telephone). Being a voice-driven system, it extends the access to ICTs (Information and Communication Technologies) to illiterate users. Many users in the developing world are now creating and accessing information through the Spoken Web platform. The content ranges from information about people and events in a village [3] to information about the crop prices in the market [2], to business information for the unorganized workforce in developing regions [20].

Once the Spoken Web enables large amount of content creation and access, the problems of information management need to be solved to ensure users are able to access the information they need. In the WWW history, the initial years of information management was performed by the directories that would create a catalog of web pages [9]. When the number of web pages kept on increasing, the catalog was not a scalable solution for content management. The

search engines then [7] changed the Internet information access paradigm, reducing the information access problem to a simple query interface. The success of the WWW owes a lot to the search engines.

As Spoken Web services reach out to individuals – for scenarios such as daily wage earners creating their VoiceSites for seeking jobs [20], or people creating VoiceSites for social networking – searching across VoiceSites will become an as much required technology as searching across the entire web is today.

It is therefore critical to the success of Spoken Web that a search interface exists which can crawl the VoiceSites, index them and then present a query-search interface to the user over phone. A user should be able to speak a query terms and the system should play back the search results, while providing an option to the user to connect to the VoiceSite that has the content — in a manner similar to the WWW search interfaces, albeit on audio. This is an important problem area from an impact perspective. It can enable information access to the three billion world population that currently does not have access to the Internet, but is using mobile devices and the voice modality for interaction.

1.1 Solution Approach

Our approach for audio content search is focused on the audio data that is created by users over the phone — akin to the user generated text content that people create in the Web 2.0 incarnation of the Internet. The audio data is in several languages and dialects and is often spontaneous audio content. If we look at standard speech recognition approaches to convert audio into text for indexing in a search system, this is not a practical approach given the quality of audio. A usual approach in situations when the actual content cannot be processed is to index the meta-data that is associated with the content. While meta-data is easy to extract from the content and is easy to index, it cannot be expected to have a complete representation of the content.

In this paper, we therefore propose using both, the meta-data and the audio content data in a novel manner. We create two separate indexes — the meta-data index has sparse representation of the data, but gives precise results on the indexed terms. We perform a low-threshold speech recognition (and audio feature recognition) to index the audio content. The correctness of the later technique is not high, but we get a large number of results — and hence a high recall. Such two-stream processing provides with a improved precision-recall results of the audio content. The audio query is passed to the two indexes and the results are then merged using two different techniques based on the confidence scores of the two result sets.

Additionally, this paper presents the engineering aspects of crawling VoiceSites, which is challenging since it differs significantly from crawling of the Web pages. We also present the audio query-search interface and the techniques that are used to transfer a user from the search site to the VoiceSite that contains the content. Together with this engineering solution, and the two-stream technical approach, we claim to provide an end-to-end search system for audio content on the Spoken Web.

1.2 Key Contributions

Given the above setting and the approach, this paper presents a two-stream indexing of the audio content on Spoken Web to build a complete search system. We use data from live settings and present the precision recall improvements by using this technique. Specifically, following are the key contributions of the work presented in this paper:

- We propose a technique for crawling the VoiceSite so that it can extract the meta-data and the corresponding audio content with respect to specific sections on the VoiceSites.
- We propose a mechanism to perform a two-stream processing of the audio content to generate two separate indexes, one for meta-data and one for the audio content features.
- We build a complete system that consists of a query-search interface and the mechanism to transfer the audio content to the specific portion in the VoiceSite, where the content lies.
- We present a mechanism to acoustically highlight the keywords in the audio content that were part of the query terms

The paper organization follows a standard structure of first presenting the related research in the field (Section 2). The Spoken Web forms a specific background section of the Section 2, since the entire content data is based on that. We then directly go into describing the data set (Section 3) since we believe that the techniques can be better understood once the reader has an idea of the data. Then we present the technique for crawling the VoiceSites (Section 4). This is followed by the description of the indexing technique (Section 5) that is presented in two parts of meta-data and the audio-content index. We present two specific approaches for the audio content index, one focusing on extracting words from the audio and the other on extracting the phonetic lattices from the audio. We then present the mechanism of presenting the results and the changes that need to be made in the index to support anchor transfers in VoiceSites (Section 6). Finally, we present the experiments and the results that illustrate the usefulness of the proposed approach (Section 7). We conclude the paper by an interesting discussions focusing on the implication of the technique in the developing world (Section 8).

2. RELATED WORK

The problem presented in this paper and the approach have been motivated by some pioneering work in the area of spoken document retrieval, crawling techniques and content metadata search areas. In this section, we present the key publications in these areas and highlight the differences from our work and also present the work on which we have built the techniques. At the end, we present the background on Spoken Web which will provide the context for the data set presented in the following section.

2.1 Audio Search and Browsing

Searching audio content is usually dependent on converting the audio to text using a speech recognizer and then performing the standard text based search. However the accuracy of speech recognizer is not very high for spontaneous conversational speech. In [31], the authors address this specific problem and suggest a very innovative way to counter

for the low accuracy of the speech recognizer. They suggest use of dichotic stream (one audio stream for one ear) to play back the search results. However this cannot be applied to a phone channel audio, which is not stereo. Alternate mechanisms to improve speech recognition in resource-constraint languages involves the concept of people participation to gather data for speech recognition [23]. Such techniques present a glimmer of hope for speech recognition in multiple languages.

Over the last five years, researchers have started to look at searching speech not through speech recognition, but by indexing speech at a subword level [38] [12]. Such techniques are more robust to speech recognition errors since they consider multiple recognition hypothesis in the index [13]. *Future research in our proposed solution should use such language independent indexing to augment content search with the metadata search presented in this paper.*

Most systems for audio search use a visual query interface for accessing content indexed from audio. This is typical of a web based search system for searching video content [4], and in call centers where supervisors wish to monitor offline content of the call center agents [27]. In such situations, the query interface is still visual and hence simple. *What differentiates our problem is that the query interface and the content, both are in audio.*

2.2 Search Interfaces on Mobile Devices

Due to the limited screen size on mobile devices, the problem of efficiently utilizing the screen for presenting search results becomes interesting. A common technique is to categorize the search results and present just the categories at a high level [18, 16]. Quicklinks [10] is another technique that provides more than just the website, but also the structure of the site so that a user can directly reach a particular portion of the site. Research has also addressed issues related to reaching a specific anchor text in a large hypertext document [19].

However since audio has not been earlier presented in a Web structure, handling anchors in a audio VoiceSite has not been addressed earlier. Owing to the linear structure of audio presentation, this is not a trivial extension from the text domain.

2.3 Metadata for Search

With the increasing ease of tagging content, and the availability of a large number of sensors, the amount of metadata associated with the content has been on the rise in the last decade. Swoogle uses metadata to define relationships between documents in the Semantic Web [15]. Metadata has been the backbone for information retrieval in Digital Libraries [6] [17]. Since it is difficult to process and extract semantics from images, metadata has been the key for image search systems [37]. Not surprisingly, the metadata based content management systems for mobiles have started to appear in the research community. The authors in [32] have used metadata to manage the images on a mobile. Reusability of mobile multimedia objects is addressed through metadata in [36].

We presented use of metadata information in Spoken Web and illustrated the use of faceted browsing of search results for Spoken Web in [14]. Other than this, *surprisingly not much work has been done on improving the audio search using metadata. This is the other differentiating space where our paper borrows ideas from image and multimedia metadata research and applies them to the audio domain.*

2.4 Spoken Web Background

The aim here is to briefly introduce the Spoken Web and refer to the papers for a detailed description of the technology. It is important to explain the Spoken Web to (a) provide an understanding of the data that is used for search in the next few sections, and (b) explain the broader applicability of the search technique proposed in this paper.

VoiceSites are voice-driven applications that are hosted in the telecom network [21]. They are addressed by a unique phone number and can be accessed from any phone instrument, mobile or land-line through an ordinary phone call to that number. The phone does not require any extra features or software to be installed on the device. VoiceSites are therefore analogous to websites in the WWW but can be accessed by dialing a phone number and information can be *heard* rather than being read or seen.

Creation of content on a visited is made easy by the VoiGen system [21] to which anyone can call and interact with it through voice. This can enable any illiterate person to create her VoiceSite. Such a system enables easy local audio-content creation. By answering simple questions as the ones shown in this interaction, an illiterate person can also create content on the VoiceSite. A VoiceSite can link to other VoiceSites through Hyperspeech Transfer Protocol (HSTP) [1]. Such interconnected VoiceSites result in a network which we refer as the Spoken Web or World Wide Telecom Web (WWTW) [22], shown in Figure 1.

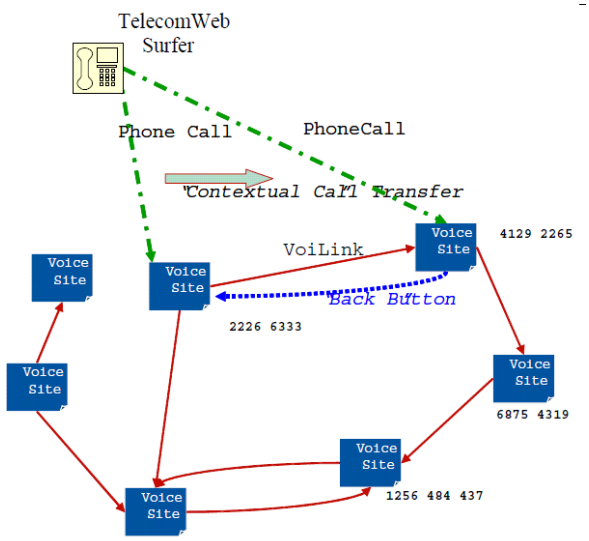

Figure 1: The World Wide Telecom Web.

System-level technology research on WWTW has been presented in [21] where the authors present technologies for VoiceSite creation. The Hyperspeech Transfer Protocol has been presented in [1] that enables linking of one VoiceSite with the other in the Telecom Web. A solution based on these technologies [20] proposes to organize the unorganized urban micro-businesses. As mentioned earlier, the focus of work presented in this paper is to enable searching of the above mentioned user-generated content.

3. DATA SETS

The data for performing the experiments is extracted by crawling seven VoiceSites. Three of these VoiceSites are from live VoiceSites that contain data created by more than 10,000 end-users [2]. The remaining four VoiceSites were demo sites, that primarily had data from the VoiceSite creator.

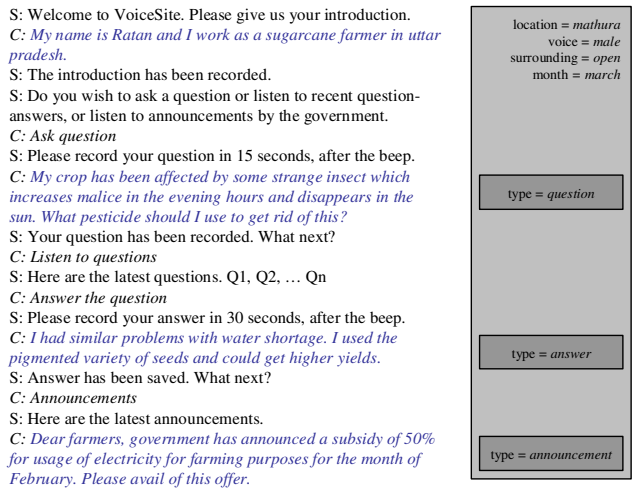

Figure 2: A sample interaction of content creation on Spoken Web. The labels on the right represents the metadata associated with the audio snippets.

A sample interaction for creating content on the VoiceSite is shown in Figure 2. The text in blue color is recorded as audio. The tags on the right show the corresponding metadata associated with the audio. The metadata consists of a variety of description about the text ranging from the location to the type of content and the time when the content was created.

Table 1 shows the amount of data that we had from each VoiceSite and the different languages that it contained the audio in. It also shows the number of users who had accessed these VoiceSites — as an indication of the popularity of the content among the community for this the VoiceSite was developed.

Table 1: Data from the 7 VoiceSites

VoiceSite	Language	Audio	Users
Mishri	Hindi	7983	1221
AvajOtlo	Gujarati	4041	127
VillagePortal	Telugu	9421	6500
MobieBazaar	Gujarati	142	32
BoxOffice	English	51	9
Grievance	Hindi	34	2
Insurance	Hindi	26	1

A total of 21698 audio documents were indexed for the purpose of the experiment. Each audio is an 8 KHz signal that has been created by the user over a phone interaction. The metadata for each document was not in a specific structure, since it came from different contexts in one of the above mentioned VoiceSites. On an average, we had about three metadata values for each audio file, with the actual value ranging from one to six values. The speech recognition extraction process is explained in Section 5.1.

4. CRAWLING OF THE SPOKEN WEB

The first step in building a search engine for Spoken Web is to crawl the VoiceSites. VoiceSites are deployed on an application server and each VoiceSite has an associated phone number that acts as a URL to the site. The Spoken Web crawler runs from a basic seed set and crawls the entire content from this set. The basic seed set is a tuple of the form $(number, site - details)$ where $number$ is the phone number of the VoiceSite and $site - details$ is an XML file that contains the URL of the VoiceSite, in addition to other detail as illustrated in a sample XML shown in Figure 3.

```
<XMLPage TYPE="DID_URL_MAP" VERSION="1.6" HREF="$start-call-url$">
        <SET VARNAME="$appxmlversion$" VALUE="2.0" />
        .
        .
        .
        <SET VARNAME="$ivr-root-dir$" VALUE= "http://192.168.10.72:8080/DSC_Radio_Navigation" />
        <SET VARNAME="$ivr-url$" VALUE="$ivr-root-url$/welcome.vxml" />
        .
        .
        .
</XMLPage>
```

Figure 3: A sample XML that consists of the VoiceSite address.

The crawler then extracts the VoiceSite application (usually authored in VoiceXML, and some associated JavaScript snippets). The crawler is similar to any Web crawler, such as WebSPHINX [26] or JSpider [35] except for the fact that it crawls for VoiceXML tags and extracts relevant information from prompts, grammars and meta-information. A typical VoiceXML fragment and the different information that is crawled is explained in Figure 4.

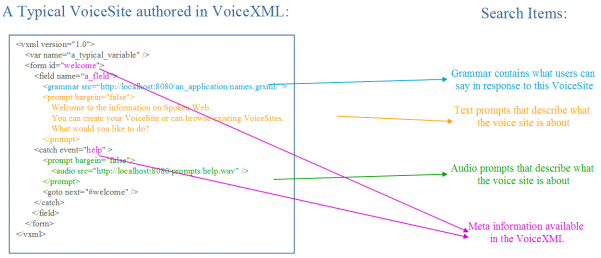

Figure 4: A sample VoiceXML snippet in a VoiceSite, and information that is crawled.

The crawler parses the text of the file, line by line until it finds tags like $< prompt >$, $< log >$, $<! - -$, $< audio$, $< submit$ and on finding these tags, it processes them as required. The main function of the crawler is to create the index which are then queried later for searching and also to extract links to continue the crawling process. Currently we have the source to the VoiceXML content of a VoiceSite, but eventually for a crawler to work on VoiceSites authored by other organizations, a standard API will have to be built that can be used to access data available for building an index. Calling a VoiceSite and getting audio data may not be of as much help because audio information will be difficult to parse and so it is not easy for crawlers to proceed in that fashion.

We create two indexes; one is the *Forward Index* for the information that is present in the JavaScript. This includes the Logs, Prompts and the Comments. The second is the *Inverted Index*, for the content that is linked to from these JavaScripts. This consists of audio files which are linked with the <audio> tag. More information on the two differ-

ent types of indexes can be accessed at [8]. The summary is that when the ratio of the number of words per document to the number of documents is high, then the *Inverted Index* is more efficient structure of the index. In our case, the number of audio files that a JavaScript can link to is often of the order of 12000, and an audio file is generally linked to by at most ten JavaScript snippets, we use the *Inverted Index* .

Next, the indexing is split into two, for the Forward and Inverted indexes. Let us first take the case of *Forward Index*. When the crawler parses the text to find the $< log >$, $<!--$, $< prompt >$ tags, then it takes the text that is stored in these tags, and indexes them in the *Forward Index*.

The mechanism for the *Inverted Index* is more complex. When the $< audio$ tag is found, then we check the kind of audio features that exist for the content. There are cases when the name of the file is not actually specified, but is provided by a session variable or an argument which is passed to the file. The values of these variables are present only when the JavaScript is being executed. This poses a serious challenge since the value of these variables are dynamic, and can be accessed only while the file is being executed, but the crawling is done when these files are not being executed, and so these values cannot be accessed. We use regular expressions to extract the audio files corresponding to such problematic scripts. One type of audio feature is the text transcripts, while the other is a phonetic transcription, which is stored separately as an index as mentioned in Section 5.1. The crawler then proceeds to find the metadata for this particular audio. The metadata is stored in XML format, and contains fields like Name, Village, Type, Month etc, which are again parsed with the help of the Digester Library, and stored in the text file.

5. TWO STREAM INDEXING

In this section, we will provide details about the two stream indexing process. The process consists of extracting the metadata associated with the audio content, and to process the audio content to extract information. As shown in Figure 5, the same source of VoiceSites is passed through two separate processes to extract the two streams. Each VoiceSite file is crawled to extract the audio file associated with the file. Then the audio file is indexed with respect to the two streams.

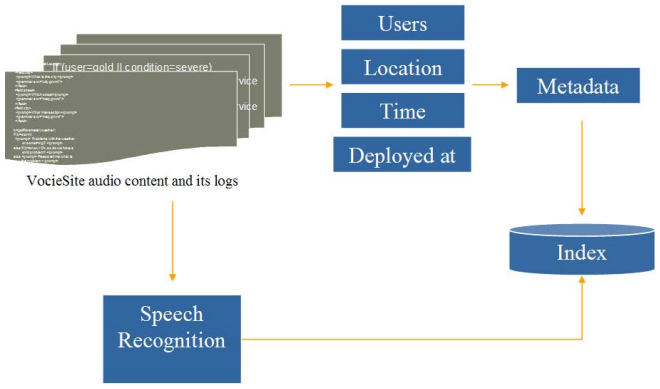

Figure 5: Two stream processing for indexing content on the Voice Sites.

The forward and inverted indexes are separately created as mentioned in the previous section. The structure of a document in a forward-index is shown in Figure 6 (a) while the inverted index structure is in Figure 6 (b). It is to be noted that not all metadata tags in Figure 6 (b) may be present for each audio file.

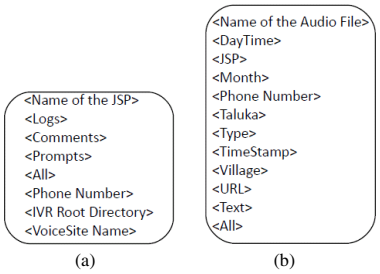

Figure 6: (a) Structure of a forward-indexed document, (b) Structure of a inverted-indexed document.

In both the indexes, all fields are searched. The forward index contains information from the the *Logs, Prompts* and *Comments*, which are all appended together. This information is extracted from the VoiceXML as was earlier shown in Figure 4. Usually a single VoiceXML file will have specific logs, prompts and comments, we use a forward index for this purpose. The Inverted Index holds the metadata fields and the text fields. The derived metadata can point to a multiple number of audio documents, and therefore we use an inverted index for this data. Eventually, all fields are combined into one to simplify searching.

We used Lucene [24] search engine library to generate the indexes. Specifically, we used the *Lucene Highlighter* to search the forward index. This returns the text corresponding to a query. The text result is then played out to the user by performing a text-to-speech synthesis.

5.1 Speech processing for indexing the audio content

We followed two specific processes for extracting information from the audio content. One is a plain vanilla speech recognition. We used the Nuance speech recognition toolkit with Indian Language acoustic models. However the Nuance acoustic models are not trained for continuous sentences, so we created separate grammars to handle the continuous sentences. The grammar was built from a text corpus of more than 30,000 words. The grammar had 7597 unique words and consists of 7597 unigrams and more than 4589 bigrams. We refrained from training a trigram grammar since we did not want to constrain the language syntax on the data that was more spontaneous in nature.

This speech recognition based information extraction approach has some limitations. First, this approach would fail when the search query issued by a user is not present in the vocabulary of the recognizer. This would hurt search performance when queries involve emerging topics or named entities. The second problem is that this approach is language dependent. A large fraction of spoken web content is likely to be in regional languages and dialects and getting speech recognizers for all these languages may not be easy.

A significant research work has been reported in literature to address these limitations [39, 28]. We build upon the ap-

proach presented in [28] where the spoken content is indexed by a finite-state-transducer (FST) which maps a phonetic substring to a set of finite-state-automata (FSA). Each FSA in this set represents a phonetic lattice corresponding to a spoken utterance. More detains on the general indexation algorithm can be found in [5]. We used open-source OpenFST toolkit [1] to implement this solution.

While the approach proposed in [28] takes account of the out-of-vocabulary problem mentioned above, it still works in language-dependent fashion. The phonetic lattices in [28] are derived from word-lattices and finally converted into an FST index. In order to make the approach adaptable to other similar languages, we perform a phonetic recognition over spoken utterances and convert the resulting phonetic lattices into FST index. Phonetic recognition requires phonetic language model as well as an acoustic model. While we train phonetic language model for every new language from corresponding text corpus, the acoustic model can be trained once and shared across all other acoustically similar languages. We applied this approach to index Gujarati spoken data using Indian-English acoustic models.

The context and speaker dependent Indian-English acoustic models were trained using 180 hours of audio data with transcripts. This data comes from approx. 1500 different speakers. In total, there are 100k utterances used for training and the average length of each utterance is approximately 6 seconds. For a set of 15 search queries about crops and pesticides names, we obtained a P@10 accuracy (fraction of top 10 search results for every query that is correct) of 0.6.

5.2 Merging the two streams

The forward and the inverse index are merged in a single index of the Lucene library. We force the forward index to hold with its structure since this provides faster access to content as mentioned earlier. In both the indexes, the content in $< All >$ is searched. In the forward index, this field contains the logs, prompts, comments, which are all appended together. In the inverted index, the metadata fields and the text fields are appended. This is done to simplify the search process. An alternative is to treat each element separately and then use a Multisearcher feature to search across fields. Thus we assign equal weight to the metadata and the audio-content search indexes. The combined scores from each of the two streams is then used to create a single search result list.

Each search result consists of four components: (a) the phone number of the VoiceSite, (b) a flag that determines if the result is an audio file or a text result, (c) the name of the particular source file in the VoiceSite that contains the search result, and, (d) the matching text or the name of the audio file where the result was present. All the results are put in a Hash Map and are then presented to the query-results interface in a sequence.

6. RESULTS PRESENTATION

As shown in Table 1, a single VoiceSite can have as many as 7000 audio documents. This presents an interesting problem in the query-search interface. When search results are presented to the users, and if a user selects a particular result to reach the VoiceSite that contains the content, then perhaps transferring the user to that VoiceSite may not be too helpful since the user would not be able to find where in the site is the content that was shown in the search result. This problem is illustrated in Figure 7 which shows that the audio that contains a search term *kapaas* could be deep in the VoiceSite.

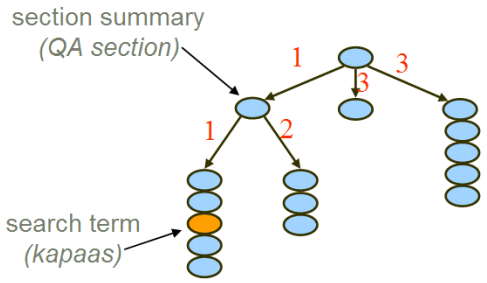

Figure 7: The searched content could be deep in a VoiceSite tree structure.

If a user is transferred to the top of the VoiceSite, then he would have to press 1 two times and then wait for the third audio file to get to the result. This is not a practical interface to transfer a user from the results page to the VoiceSite. If a search term exists down below in a particular leg, then it is important to transfer to that leg else a user will never realize where to navigate to get to the searched content. Reaching this content requires user-interactions, which is not available if we wish to do an automatic transfer. Further, a summary of where the user is in the tree, will also be required to ease the navigation.

To overcome these challenges, we developed a method for transferring the control to a specific section in a VoiceSite by identifying the section that contains the search result in the actual VoiceSite and by providing the ability to transfer to a particular section in the VoiceSite. We also developed a method for identifying, and then marking, and then dynamically playing the marked audio that contains the search term for better usability.

For this purpose, we associate a landing information with every audio content of the VoiceSite. The landing information contains information that consists of specific section names and the session variables that are required to be set to reach the page automatically. A sample landing information is shown in Figure 8.

```
<% landing_name = "LEG-1";%>
<% landing_var1 = "sec_name"; %>
<% landing_val1="1";%>
<% landing_var2 = "sub_sec_name"; %>
<% landing_val2="1";%>
<% landing_iteration=3; %>
```

Figure 8: The landing information that helps in transferring to a specific portion in the VoiceSite.

As a result of this landing information, the index is modified to contain the landing information in addition to the metadata and the audio data. When the search results are presented to the user, each result is additionally appended by the landing information and the time where the query

[1] www.openfst.org/

appears in the audio, in addition to the phone number of the VoiceSite to be transferred to, as shown in Figure 9.

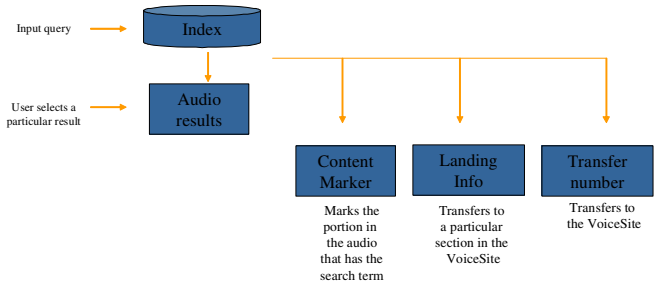

Figure 9: Additional information that is appended to the search result – to enable transfer to specific portion in the VoiceSite.

The transfer number helps in transferring the call to the VoiceSite number. The landing information helps in identifying the right section in the VoiceSite. The timing location of the query term is used to mark the content as explained next.

In a visual text-based search interface, these days when we search for a query term, then the browser highlights the query terms in the visual display. To emulate this feature, we use the timing information from the index to identify the location of the search term in the result set. We then either append a signature music or increase the energy in the audio for that specific portion of the audio that has the query term. This provides a clear indication to the user that his search term appears in the content. Figure 10 is an illustration of this feature.

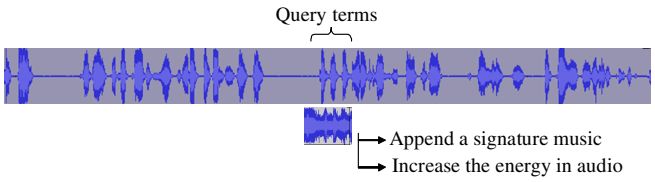

Figure 10: A mechanism to highlight the query terms in the search results.

The concept of indexing audio content by performing speech processing and by extracting the metadata information, and then the concept of providing a smooth transfer from a query-results interface to a VoiceSite, we demonstrate a highly usable search system for the Spoken Web.

7. RESULTS AND DISCUSSIONS

The evaluation of the two-stream audio search system presented in this paper focuses on the multiple-languages of audio content and the effectiveness of the two-stream process as opposed to using just either of the single stream.

7.1 Query Terms

The results are derived from a set of 100 query terms that were created to contain terms across the seven VoiceSites. Since the search system is currently not available to end users, it is difficult to determine the kind of query terms that people would be interested in searching. So the terms presented in this section are the ones that the authors perceive to be of value. The transliterated query terms are shown in Figure 11.

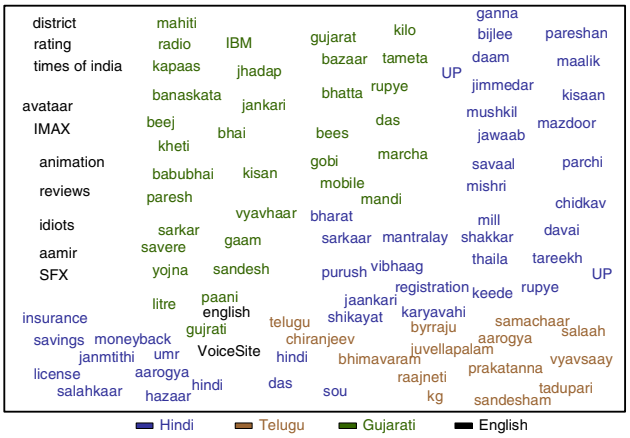

Figure 11: The 100 words that were used as query terms for the experiment.

The color of the terms differentiate between the different languages of the search terms. All these query terms were spoken to create the audio of each term. In all, there are 100 words, of which 12 words from English, 34 words from Gujarati, 14 words from Telugu and 40 words from Hindi language. The mix of words across languages were chosen to evaluate the effectiveness of the two-stream search process across multiple-languages.

7.2 VoiceSite Search Results

The first aim is to identify whether a metadata system or a speech recognition system or the two-stream system could get the correct VoiceSite in its result set.

Table 2: Precision rates for VoiceSite number task for top-3 and top-5 search results.

Index Type	Precision (@3)	Precision (@5)
Metadata	94.8%	96.2%
Speech Features	51.2%	55.2%
Two stream	80.0%	82.4%

Table 2 presents the success percentage across the three different indexes, while considering the top three search results (P@3) and the top five search results (P@5). It is clear that the two stream processing system results in a higher accuracy of extracting the VoiceSite, albeit by a smaller fraction. Since reaching to the VoiceSite involves the information about the language and other metadata of a particular VoiceSite, the precision is very high for the metadata based search system. Moreover, some of the information such as the name of the VoiceSite (query terms such as *gujrati, telgu, hindi* which are the language of the VoiceSite) is mostly contained in the metadata and is missing from the audio content of a VoiceSite, therefore the speech recognition based indexing does not yield good results. When the two-stream processing system uses inputs from the audio content indexing, the precision suffers slightly, especially at low values of recall as suggested in Table 2. The increased number of search results, from three to five, does not yield a

significant improvement in the precision. We owe this saturation to the fact that since there are only seven VoiceSites in total, if the content is not found in the metadata, then it won't show up the correct VoiceSite even if the number of search results are increased.

Figure 12 shows the precision-recall graph for the above experiment. Since we do not have a large amount of metadata content, we do not get a large number of search results in the metadata index. Therefore the graph has missing points on high value of recall for the metadata part of the curve. Ideally the curve should reach out to a recall value of one when all the possible content are returned by the system – thus reducing the precision significantly.

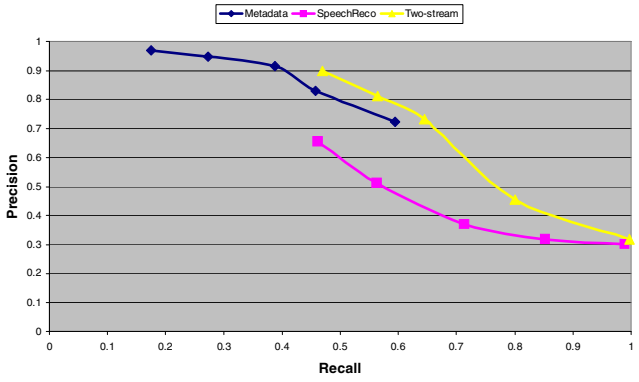

Figure 12: The precision-recall curve for the VoiceSite extraction task.

The tapering of the two-stream index curve at high value of recall is a reflection of the missing metadata information at those recall values. When a large number of search results are fetched, then most of them are due to the audio content, as the amount of metadata is limited, and therefore the precision of the two-stream system reduces significantly at higher values of recall. Considering the fact that we are able to get about 80% of the correct results within 3 search results is a significant result, given the fact that the query terms are across four languages.

7.3 Content Search Results

The next experiment was to identify if the search result is able to retrieve the specific audio content that is relevant to the query term. This is a relatively tough task since the number of VoiceSites are seven, but the number of audio documents are far too large (20798, as shown in Table 1). The same search terms mentioned in Figure 11 are used for this experiment. The precision-recall curve for the three techniques is shown in Figure 13.

Since the search for a specific content in the VoiceSite uses the audio content information heavily as compared to the VoiceSite search task mentioned earlier, the improvements in using the audio-content along with the metadata are obvious as shown in the figure. Except for the very low recall values, the precision of the two-stream processing engine is very high. For low values of recall, the precision for the metadata system is very high since it either has the query term or not, with less error. So if the query term is present in the metadata, then metadata based search system is obviously able to retrieve it. However at higher values of recall, the metadata system does not have sufficient data to pull out

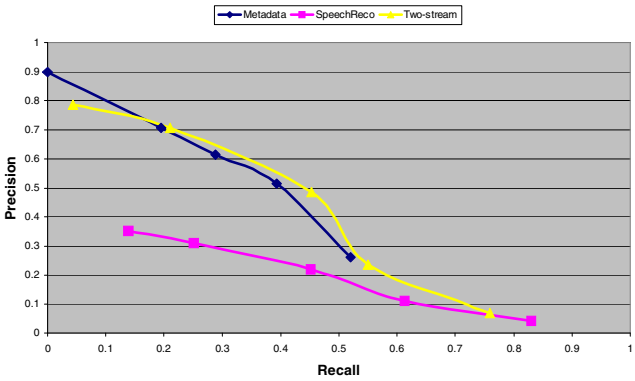

Figure 13: The precision-recall curve for the content extraction task.

the corresponding audio content and so its accuracy drops significantly as shown in 13. On the other hand, the two-stream index is able to get more relevant search results from the audio content analysis and so its accuracy does not go as low as the metadata index system. However the two-stream search system does suffer at low values of recall, when the audio content search results tend to increase the noise in the otherwise clean metadata index. This is perhaps a trade-off of the two-stream search system.

At a reasonable recall level, however, it is very clear that the two stream processing performs far better than either of the processing technique. The few search results in the metadata is a result of the limited metadata that exists for some audio documents. This is in fact the limitation of a metadata-only approach.

8. CONCLUSION AND FUTURE WORK

This paper presented a technique for two stream indexing of audio content on the Spoken Web. We presented an end-to-end system that includes a crawler, the indexer and the results presentation system. The design is influenced by the specific nuances of the audio data and the limitations of the audio interaction process. The two stream indexing is supposed to provide better search results since the two streams separately focus on the precision and recall. The results presentation is meant to be a more easier and practical user interface especially for VoiceSites that contain a large amount of audio content. The results provide the importance of the two stream search approach. The results also highlight the effectiveness of the system to index more than just the audio, but also the location of the audio in the file. Being a first search system for Spoken Web, the results are encouraging.

In the future we intend to perform the missing piece of a usability study to evaluate the effectiveness of the system from the users' perspective. The study will evaluate the user-interface (especially, the concept of highlighting the query term) and also the appropriateness of search results for the users. We would also like to work on extending the concepts of phonetic lattices for audio content indexing, as briefly mentioned in Section 5.1. The expectation is that a phonetic index can be created for audio content across different languages. The early results as mentioned in Section 5.1 are encouraging. If the audio content search could

be improved, it is likely to improve the two-stream search results as well.

The research challenges in Spoken Web search are significantly many (we have not even discussed the ranking aspects) and this paper hopes to present the importance, initial approach and a platform — interesting enough for other researchers to take this work forward in the future.

9. REFERENCES

[1] S. Agarwal, D. Chakraborty, A. Kumar, A. A. Nanavati, and N. Rajput. HSTP: Hyperspeech Transfer Protocol. In *ACM Hypertext 2007*, UK, September 2007.

[2] S. Agarwal, K. Dhanesha, A. Jain, A. Kumar, S. Menon, N. Rajput, K. Srivastava, and S. Srivastava. Organizational, social and executional implications in delivering ict solutions: A telecom web case-study. In *Proc. Intl. Conf. on Information and Communication Technologies and Development (ICTD)*, 2010.

[3] S. Agarwal, A. Kumar, A. A. Nanavati, and N. Rajput. Content creation and dissemination by-and-for users in rural areas. In *Proc. Intl. Conf. on Information and Communication Technologies and Development (ICTD)*, April 2009.

[4] C. Alberti, M. Bacchiani, A. Bezman, C. Chelba, A. Drofa, H. Liao, P. Moreno, T. Power, A. Sahuguet, M. Shugrina, and O. Siohan. An audio indexing system for election video material. In *In Proc. ICASSP*, April 2009.

[5] C. Allauzen, M. Mohri, and M. Saraclar. General indexation of weighted automata - application to spoken utterance retrieval. In *Proceedings of the Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval at HLT/NAACL*, pages 33–40, 2004.

[6] M. Baldonado, C. chuan K. Chang, L. Gravano, and A. Paepcke. The stanford digital library metadata architecture. *International Journal of Digital Libraries*, 1:108–121, 1997.

[7] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1):107–117, 1998.

[8] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *The International Journal of Computer and Telecommunications Networking*, 1(6), 2000.

[9] A. Callerya and D. Tracy-Proulxa. Yahoo! cataloging the web. *Journal of Library Metadata*, 1(1), 1997.

[10] D. Chakrabarti, R. Kumar, and K. Punera. Quicklink selectoin for navigational query results. In *WWW '09: Proceedings of the 18th international conference on World Wide Web*, Madrid, Spain, May 2009.

[11] J. Charzinski. Traffic Properties, Client Side Cachability and CDN Usage of Popular Web Sites. *Lecture Notes in Computer Science*, 2010(5987), 2010.

[12] C. Chelba and A. Acero. Position specific posterior lattices for indexing speech. In *ACL '05: Proceedings of the Annual Meeting on Association for Computational Linguistics*, pages 443–450, 2005.

[13] T. K. Chia, K. C. Sim, H. Li, and H. T. Ng. A lattice-based approach to query-by-example spoken document retrieval. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 363–370, 2008.

[14] M. Diao, S. Mukherjea, N. Rajput, and K. Srivastava. Faceted search and browsing of audio content on spoken web. In *CIKM '10: Proceedings of the nineteenth international conference on Information and knowledge management*, 2010.

[15] L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs. Swoogle: a search and metadata engine for the semantic web. In *CIKM '04: Proceedings of the thirteenth international conference on Information and knowledge management*, pages 652–659, 2004.

[16] T. Heimonen and M. Kaki. Mobile finder: supporting mobile web search with automatic result categories. In *Proceedings of the MobileHCI*, 2007.

[17] B. Hughes and A. Kamat. A metadata search engine for digital language archives. *Digital Libraries Magazine*, 11(2), 2005.

[18] M. Jones, G. Buchanan, and H. Thimnbleby. Improving web search on small screen devices. *Interacting with Computers*, 4(15), 2003.

[19] R. Kraft and J. Zien. Mining anchor text for query refinement. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, New York, USA, May 2004.

[20] A. Kumar, N. Rajput, S. Agarwal, D. Chakraborty, and A. A. Nanavati. Organizing the unorganized – employing it to empower the under-privileged. In *Proceedings of the World Wide Web*, April 2008.

[21] A. Kumar, N. Rajput, D. Chakraborty, S. Agarwal, and A. A. Nanavati. Voiserv: Creation and delivery of converged services through voice for emerging economies. In *WoWMoM'07 Proceedings of the 2007 International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Finland, June 2007.

[22] A. Kumar, N. Rajput, D. Chakraborty, S. Agarwal, and A. A. Nanavati. WWTW: A World Wide Telecom Web for Developing Regions. In *ACM SIGCOMM Workshop on Networked Systems For Developing Regions*, Aug 2007.

[23] J. Ledlie, B. Odero, E. Minkov, I. Kiss, and J. Polifroni. Crowd translator: On building localized speech recognizers through micropayments. *SIGOPS Operating Systems Review*, 43(4), 2009.

[24] M. McCandless, E. Hatcher, and O. Gospodneti. *Lucene in Action, Second Edition*. Manning Publications Company, 2008.

[25] I. Medhi, A. Sagar, and K. Toyama. Text-Free User Interfaces for Illiterate and Semi-Literate Users. In *ICTD*, Berkeley, USA, May 2006.

[26] R. Miller and K. Bharat. Sphinx: A framework for creating personal, site-specific web crawlers. In *WWW '98: Proceedings of the 7th international conference on World Wide Web*, Brisbane, Australia, May 1998.

[27] G. Mishne, D. Carmel, R. Hoory, A. Roytman, and A. Soffer. Automatic analysis of call-center conversations. In *CIKM '05: Proceedings of the 14th international conference on Information and knowledge management*, pages 453–459, 2005.

[28] C. Parada, A. Sethy, and B. Ramachandran. Query-by-example spoken term detection for OOV terms. In *In Proc. ASRU*, December 2009.

[29] N. Patel, D. Chittamuru, A. Jain, P. Dave, and T. S. Parikh. Avaaj Otalo - A Field Study of an Interactive Voice Forum for Small Farmers in Rural India. In *Proc. CHI*, USA, April 2010.

[30] M. Plauch, U. Nallasamy, J. Pal, C. Wooters, and D. Ramachandran. Speech Recognition for Illiterate Access Information and Technology. In *ICTD)*, Berkeley, CA, USA, May 2006.

[31] A. Ranjan, R. Balakrishnan, and M. Chignell. Searching in audio: the utility of transcripts, dichotic presentation, and time-compression. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 721–730, 2006.

[32] R. Sarvas, E. Herrarte, A. Wilhelm, and M. Davis. Metadata creation system for mobile images. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 36–48, 2004.

[33] U. N. E. Scientific and C. Organization. Education for All Global Monitoring Report - Reaching the Marginalized. *http://unesdoc.unesco.org/images/0018/001866/186606E.pdf*, pages 16–32, 2010.

[34] J. Sherwani. Are Spoken Dialog Systems Viable for Under-served Semi-literate Populations? *PhD Thesis Proposal, Carnegie Mellon University, http://www.cs.cmu.edu/ jsherwan/JS-proposal.pdf*, 2005.

[35] Sourceforge. Jspider - the Open Source Web Robot. *http://j-spider.sourceforge.net*, October 2010.

[36] M. Svensson and A. Kurti. Using contextual metadata for enhanced reusability of mobile media objects. In *Sharing Experiences with Social Mobile Media : Proceedings of the International Workshop in conjunction with MobileHCI*, pages 72–79, 2009.

[37] K.-P. Yee, K. Swearingen, L. Li, and M. Hearst. Faceted metadata for image search and browsing. In *CHI '03: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 401–408, 2003.

[38] K. C. Yu, C. Ma, and F. Seide. Vocabulary independent indexing of spontaneous speech. *IEEE Transactions on Speech and Audio Processing*, 13(5), 2005.

[39] Y. Zhang and J. Glass. Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams. In *In Proc. ASRU*, December 2009.