Babler - Data Collection from the Web to Support Speech Recognition and Keyword Search

Gideon Mendels **Erica Cooper** **Julia Hirschberg**
Columbia University, New York, USA
gm2597@columbia.edu {ecooper, julia}@cs.columbia.edu

Abstract

We describe a system to collect web data for Low Resource Languages, to augment language model training data for Automatic Speech Recognition (ASR) and keyword search by reducing the Out-of-Vocabulary (OOV) rates – words in the test set that did not appear in the training set for ASR. We test this system on seven Low Resource Languages from the IARPA Babel Program: Paraguayan Guarani, Igbo, Amharic, Halh Mongolian, Javanese, Pashto, and Dholuo. The success of our system compared with other web collection systems is due to the targeted collection sources (blogs, twitter, forums) and the inclusion of a separate language identification component in its pipeline, which filters the data initially collected before finally saving it. Our results show a major reduction of OOV rates relative to those calculated from training corpora alone and major reductions in OOV rates calculated in terms of keywords in the training development set. We also describe differences among genres in this reduction, which vary by language but show a pronounced influence for augmentation from Twitter data for most languages.

1 Introduction

Collecting data from the web for commercial and research purposes has become a popular task, used for a wide variety of purposes in text and speech processing. However, to date, most of this data collection has been done for English and other High Resource Languages (HRLs). These languages are characterized by having extensive computational tools and large amounts of readily available web data and include languages such as French, Spanish, Mandarin, and German. Low Resource Languages (LRLs), although many are spoken by millions of people, are much less likely and much more difficult to mine, due largely to the smaller presence these languages have on the web. These include languages such as Paraguayan Guarni, Igbo, Amharic, Halh Mongolian, Javanese, Pashto, and Dholuo, inter alia.

In this paper we describe a new system which addresses the problem of collecting large amounts of LRL data from multiple web sources. Unlike current HRL collection systems, Babler provides a targeted collection pipeline for social networks and conversational style text. The purpose of this data collection is to augment the training data used by Automatic Speech Recognition (ASR) to create language models ASR and for Keyword Search (KWS) for LRLs. The more specific goal is to reduce the Out-of-Vocabulary (OOV) rates for languages when the amount of data in the training set is small and thus words in the test set may not occur in the training set. Web data can add many additional words to the ASR and KWS lexicon which is shown to improve performance over WER and KW hit rate. Critically, this web data must be in a genre close to that of the ASR training and test sets which is the main reason we developed a pipeline that focuses on conversational style text. In this paper we describe the properties which LRL web collection requires of systems, compare ours with other popular web collection and scraping software, and describe results achieved for reducing Word Error Rate (WER) for ASR and OOVs and improvements in the IARPA Babel keyword search task.

In Section 2 we describe previous research in web collection for speech recognition and keyword search. In Section 3 we briefly describe the

IARPA Babel project and we describe its language resources. In Section 4 we describe the components of our web collection systems. In Section 5 we identify the web sources we use. In Section 6 we compare our system to other tools for web data collection. In Section 7 we describe subsequent text normalization used to prepare the collection material for language modeling. In Section 8 we describe results of adding collected web data to available Babel training data in reducing OOV rates. We conclude in Section 9 and discuss future research.

2 Previous Research

A number of tools and methodologies have been proposed for web scraping use in building web corpora for speech and NLP applications. Baroni and Bernardini (2004) developed BootCat to generate search engine queries in an iterative process in order to create a corpus typically for specific domains. De Groc et al (2011) optimized the query generation process by graph modeling the relationship between queries, documents and terms. This approach improved mean precision by 25% over the BootCat method. Hoogeveen and Pauw (2011) used a similar query generation method but incorporated language identification as part of their pipeline. In text-based research, web resources have been mined by researchers to collect social media and review data for sentiment analysis ((Wang et al., 2014);(C. Argueta and Chen, 2016)), to improve language identification (Lui et al., 2014), to find interpretations of compound nominals (Nicholson and Baldwin, 2006), to find variants of proper names (Andrews et al., 2012), to provide parallel corpora for training Machine Translation engines, to develop corpora for studies of code-switching (Solorio et al., 2014), to predict chat responses in social media to facilitate response completion (Pang and Ravi, 2012), inter alia. In each case the data collected will differ depending upon the application.

However, in speech research, web data collection has been largely focused on improving ASR and KWS, where insufficient data may be available from existing training corpora. Until recently, most attempts at data augmentation from the web have been confined to HRLs such as English, French, and Mandarin. In ASR research, improved performance has been achieved by supplementing language model training data with web data in different domains (Iyer et al., 1997), particularly when that data closely matches the genre of the available training material and the task at hand (Bulyko et al., 2003). While earlier work focused on English, (Ng et al., 2005) extended this approach to the recognition of Mandarin conversational speech and Schlippe et al 2013 explored the use of web data to perform unsupervised language model adaptation for French Broadcast News using RSS feeds and Twitter data. Creutz et al. (2009) presented an efficient method for selecting queries to extract useful web text for general or user-dependent vocabularies. Most of this research has used perplexity to determine improvement resulting from the addition of web text to the original language model corpus (Bulyko et al., 2007) although (Sarikaya et al., 2005) have also proposed the use of BLEU scores in augmenting language model training data for Spoken Dialogue Systems.

In recent years, the use of web data has begun to be used to improve OOV rates for ASR and KWS performance on LRLs in the IARPA Babel project (Harper, 2011) which presents major new challenges. Web data for these languages is typically much scarcer than for HRLs, particularly in genres that are similar to the telephone conversations used in this project; since many of these LRLs are spoken with significant amounts of *code-switching*, which must be identified during web scraping, collecting data for Babel LRLs is much more complex than for other languages. Language ID is thus also an important component of LRL web data collection.

(Gandhe et al., 2013) used simple web query word seeding from the Babel lexicon on Wikipedia data, news articles and results from 30 Google queries for five of the Babel Base Period languages: Cantonese, Pashto, Tagalog, Turkish and Vietnamese. This approach improved OOV rates by up to 50% and improved Actual Term Weighted Value (ATWV) (Fiscus et al., 2007) by 0.0424 in the best case (larger values of ATWV represent improved performance), compared to a baseline system trained only on the Babel Limited Language Pack data which was provided for the task of recognition and search; each corpus consisted of ten hours of transcribed conversational speech. On average, ATWV was improved by 0.0243 across all five languages. (Zhang et al., 2015) used automatically generated query terms

followed by simple language identification techniques to reduce OOV rates for Babel Very Limited Language Packs (three hours of transcribed telephone conversations) on Cebuano, Kazakh, Kurdish, Lithuanian, Telugu and Tok Pisin. Using a variety of web genres, they managed to halve the OOV on the development set and to improve keyword spotting by an absolute 2.8 points of ATWV.

In our work, (Mendels et al., 2015), working on the same data and using a variety of additional web genres including blogs, TED talks, and online news sources obtained from keyword searches seeded by the 1000 most common words in each language, together with BBN-collected movie subtitles, all filtered by several language ID methods, we reduced OOV rates by 39-66% and improved Maximum Term Weighted Value (MTWV) by 0.0076-.0.1059 absolute points over the best language models trained without web data. In this paper, we describe an enhanced version of our system for collecting LRL data from the web, including collection of Paraguayan Guarani, Igbo, Amharic, Halh Mongolian, Javanese, Pashto, and Dholuo.

3 The Babel Program

The work presented here has been done within the context of the IARPA Babel program (Harper, 2011), which targets rapid development of speech processing technology in LRLs, focusing on keyword search in large speech corpora from ASR transcripts. The Babel program currently provides language packs for 24 languages: IARPA-babel101-v0.4c Cantonese 205b-v1.0a, 102b-v0.5a Assamese, 103b-v0.4b Bengali, 104b-v0.4a Pashto, 105b-v0.4 Turkish, 106-v0.2f Tagalog, 107b-v0.7 Vietnamese, 201b-v0.2b Haitian Creole, 202b-v1.0d Swahili, 203b-v3.1a Lao, 204b-v1.1b Tamil, 205b-v1.0a Kurmanji Kurdish, 206b-v0.1e Zulu, 207b-v1.0b Tok Pisin, 301b-v1.0b Cebuano, 302b-v1.0a Kazakh, 303b-v1.0a Telugu, 304b-v1.0b Lithuanian, 305b-v1.0b Paraguayan Guarani, 306b-v2.0c Igbo, 307b-v1.0b Amharic, 401b-v2.0b Halh Mongolian, 402b-v1.0b Javanese, and 403b-v1.0b Dholuo. We describe our system and evaluate it on the last six languages (the current phase languages) as well as Pashto. This data was collected by Appen and is released in three subsets: Full Language Packs (FLPs), consisting of 80 hours of transcribed (primarily) telephone conversations between two speakers and recorded on separate channels under a variety of recording conditions; Limited Language Packs (LLPs) with 10 hours of transcribed speech; and Very Limited Language Packs (VLLPs) with 3 hours of transcribed speech from the FLP corpus. We evaluate here on the LLP lexicons (derived from the 10 hour transcripts) for the seven languages examined. The speakers are diverse in terms of age and dialect and the gender ratio is approximately equal. A main goal of the Babel program is determining how speech recognition and keyword search technology can be developed for LRLs using increasingly smaller data sets for training. This makes data augmentation via web collection increasingly important. The major goal of the program is determining how quickly ASR and KWS systems can be developed for new languages when little transcribed speech data is initially available for use.

4 Web Data Collection

A major constraint on our data collection effort is that we must collect and process as much data as possible in a given (very short) amount of time. This constraint is designed to simulate a situation in which speech processing tools for a new language for which ASR and keyword search tools are not already available and must be created quickly. With that requirement in mind we designed a highly customizable, multi-threaded pipeline for the task (Figure 1). The pipeline consists of the following components:

1. Seeding language models
2. Search Producer
3. Job Queue
4. Scraper
5. Language identification
6. Database

We first provide an overview of the source-independent components (shown in Figure 1) and then describe in detail how we collect data from each source.

4.1 Seeding Language Models

The first component in the pipeline depicted in Figure 1 is responsible for generating keywords for seeding searches. Independent of the actual

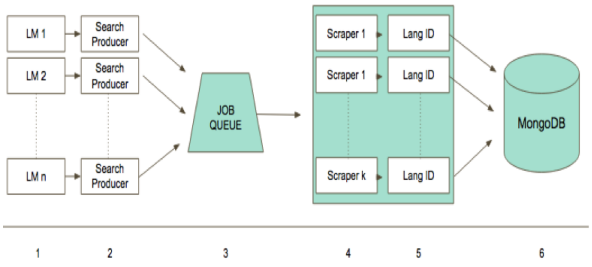

Figure 1: Data Collection Pipe Line

search provider (e.g. Bing API, Twitter API), this component is based on pre-computed unigram language models for each of the languages we want to collect. The unigram model provides the search query as explained below in Section 4.2. We compute the frequency of each token in the dataset and then remove all tokens shorter than 4 characters or tokens that occur in a standard English word list (SIL, 1999). The primary reason for removing these tokens is to reduce the number of English search results in later steps. We discovered that a query containing an English word is likely to produce mainly English results, even if that word is shared with another language, due to the heavy preponderance of English material on the web. The data for the unigram models is obtained from the Babel program; also from the Leipzig corpora (Quasthoff et al., 2006), a multilingual corpus collected from the web; and from the Crubadan project (Scannell, 2007), another multilingual corpus providing trigram counts for more than 2000 languages and dialects. Our system also supports generating bigram and trigram queries which improves accuracy of the target language results but lowers recall.

4.2 Search Production

The search production component of our systems polls a keyword from the seeding model and generates a search query. Different search providers are implemented based on the same interface to allow flexibility in adding additional search providers later. Our system currently supports Bing search API, DuckDuckGo API, Google Search, Twitter API and Topsy API.

4.3 Job Queue

The search producer described in Section 4.2 adds jobs to the queue. Each job contains the URL or data that should be inspected by the scraper. Using a concurrent blocking queue in a producer-consumer design pattern, we allow the search producer and the scraper components to work concurrently and independently, thus reducing the overhead of waiting for HTTP requests.

4.4 Scraper

This component is the heart of the pipeline and is responsible for fetching a data source, extracting the data from that source and passing it further down the pipeline.

4.5 Language Identification

Raw data that is collected is examined using our language identification multi-classifier, majority vote approach. Lui and Baldwin (2014) showed that using a majority vote over three independent language classifiers consistently outperforms any individual system, so we use the following classifiers:

- LingPipe - A language identification classifier built from LingPipe (http://alias-i.com/lingpipe/), and described in Mendels et al. (2015)
- TextCat - We implemented the TextCat algorithm (Cavnar et al., 1994) using pre-computed counts from the Crubadan Project. (Scannell, 2007)
- Google's Compact Language Detector 2 [1] - CLD2 is a Nave Bayesian classifier that supports 83 languages. We implemented a Java native interface to the original CLD2 distribution.

4.6 Database

We use MongoDB, a noSQL document-oriented database system, to store the filtered data. MongoDB allows us to process the data easily via its built in *map-reduce* component. Using MongoDB provided significant improvements compared to saving documents as text files; for example, in a single task of counting the number of tokens in the entire data set we found that MongoDB was approximately three orders of magnitude faster than using ext4 FS on Ubuntu. By overriding MongoDB internal id field we also solve the issue of duplicates, which we encounter in many sources, especially Twitter data, where tweets are often

[1] https://github.com/CLD2Owners/cld2

retweeted. To avoid saving duplicates or laboriously checking the entire dataset, we compute the SHA256 hash code for each data source and save that as the internal id field. Since this field is defined as unique over the entire MongoDB collection we avoid duplicates by definition.

5 Web Sources

5.1 Blogs - By Rich Site Summary (RSS)

RSS feeds are structured XML feeds that usually contain the latest posts from a blog. Since the data is completely structured, the task essentially involves simply fetching and parsing the XML file and extracting the correct node. We collect blog data from `blogspot.com` and `wordpress.com`. Once the search producer polls a keyword from the unigram model it constructs a Bing search query of the following form `site:blogspot.com unigram NOT lang:en`. The query consists of a domain filter, a keyword and a language filter that removes all results classified as English by Bing. The result from this query is a list of blog posts that contain the keyword. We classify the raw text using our language identifier and, if it matches the language we seek, we save the blog post.

In some cases RSS feeds are either unavailable or contain only the first paragraph of a blog post. In such cases it is necessary to separate the actual content of the post from ads, menus and other boilerplate data. To collect these posts we explored two methods for boilerplate removal:

- DiffBot, a commercial service that builds a structured representation of an HTML page by rendering it and breaking it down into its component parts using computer vision techniques.
- A pre-trained ML model (Kohlschütter et al., 2010) that uses shallow text features such as number of words and text density to separate content from boilerplate.

5.2 Forums

For web forums, we target forums created using phpBB, an open-source forum/bulletin management system. Once the search producer polls a keyword from the unigram model, it constructs a Bing search query of the following form: `Powered by phpBB AND unigram NOT lang:en`. Many phpBB forums follow the same Document Object Model (DOM) structure, for which we have written a custom scraper based on Cascading Style Sheets (CSS) style queries. Once a thread is found to be a match, we crawl the entire forum for additional threads

5.3 Twitter By Query

We poll a keyword from the unigram model and produce a search query on the Twitter and Topsy.com APIs. Both APIs are the same in terms of content but using both facilitates provides a higher throughput. The tweets in the search results are cleaned from mentions, urls, hashtags and emojis prior to language identification.

5.4 Twitter By User

An independent service revisits all the user pages from which we have collected tweets successfully in the language desired and crawls through their public history to find more tweets from the same user. This is based on the assumption that a user who tweets in a specific language will be more likely to have more tweets in that language.

5.5 TED Talks

TED.com is a website that is devoted to spreading ideas, usually in the form of short, powerful talks. Many of the talks are offered with user-translated subtitles. We use CSS queries and simple URL manipulation to download all the subtitles.

5.6 News

In some cases we have also implemented custom CSS query-based scrapers for news sites. This approach provides data with very little noise but requires implementing a manual scraper for each page.

5.7 Wikipedia

Our system also supports downloading and processing Wikipedias XML dumps, which are available for many LRLs.

6 Comparison to Other Data Collection Tools

Most tools for bootstrapping corpora-building from the web were designed for languages with a large presence in the web and for building corpora for a specific topics and terminology. Keyword search and ASR language modeling in telephone conversations collected for LRLs requires a different type of corpus. We aim to build a topic

independent, conversational corpus with very little noise in the form of HTML, JavaScript and out-of-language tokens. With this in mind, our system was designed in three main parts.

6.1 Query Generation and Sources

Topic and terminology-oriented corpora-building requires robust query generation (similar to our search producer step). It is preferable to fetch a specific subset of the documents available from the search engine. BootCat (Baroni and Bernardini, 2004) randomly generate ngram queries from the unigram seeding model. GrawlTCQ (De Groc et al., 2011) further develops the query generation process by modeling the links between documents, terms and queries. CorpusCollie (Hoogeveen and Pauw, 2011) uses a similar approach but also removes tokens that are considered to be stop-words in other languages.

Our system queries only documents from specific sources that are most suitable for our corpus: blogs, forums, twitter and subtitles rather than the entire web. This choice is dictated by the fact that the ASR language modeling and keyword search tasks that we target involve conversational telephone speech: thus, more ”conversational” text is most useful. Furthermore, when working with LRLs, we optimize the initial query generation process for recall and not precision, which explains our use of basic unigrams. Since there are very few resources available, we filter documents using language identification rather than by query design. Nonetheless we have also implemented support for bigram and trigram seeding models in cases where it would be desirable.

6.2 Language Identification and Boilerplate Removal

BootCat (Baroni and Bernardini, 2004) and GrawlITCQ (De Groc et al., 2011) have no language identification support or boilerplate removal. CorpusCollie (Hoogeveen and Pauw, 2011) uses regular expressions based filtering to remove boilerplate. For example if an HTML element contains © it is likely to be boilerplate. Rule based methods are language dependent and considered to be less robust than a machine learning models, as have been shown by Kohlschütter et al. (2010). Our system uses state of the art boilerplate removal and language identification as part of the pipeline.

6.3 Performance

Our system uses multithreading to reduce the overhead of the many HTTP requests required in web data collection. Furthermore all the tools described above use the operating system file system to manage collected documents. As shown in section 4 we have found that using a production level database system is preferable in both performance and scale.

7 Text Normalization

As previously noted, we are collecting web data for the purpose of including it in the language models for ASR that will be used to transcribe data for a spoken keyword search task. Due to the noisy nature of text found on the web, we must clean our collected data to make it appropriate for this task. Our text normalization proceeds in three distinct steps:

- Pre-normalization: a first pass in which non-standard punctuation is standardized;
- Sentence segmentation: which is accomplished using the Punkt module of NLTK (Kiss and Strunk, 2006); and
- Post normalization: in which sentence-by-sentence cleaning of any out-of-language text and standardization of numerals is done.

7.1 Pre-normalization

During pre-normalization, we first remove list entries and titles, since those generally are not full sentences. We replace non-standard characters with a standard version: these include ellipses, whitespace, hyphens, and apostrophes. Hyphens and apostrophes are removed as extraneous punctuation, except word-internal cases such as hyphenated words or contractions. Finally, any characters not part of the language's character set, the Latin character set, numerals, or allowed punctuation are removed. This cleans special characters such as symbols from the data. Latin characters are preserved, even for languages which use a different alphabet, to enable more accurate removal of entire sentences containing foreign words and URLs during post-normalization.

7.2 Sentence Segmentation

We perform sentence tokenization using the Punkt module of NLTK. Punkt uses a language-independent, unsupervised approach to sentence

boundary detection. It learns which words are abbreviations as opposed to sentence-final words, based on three criteria: First, abbreviations appear as a tight collocation of a truncated word and a final period. Second, abbreviations tend to be very short. Third, abbreviations sometimes contain internal periods. Once the abbreviations in the training corpus are learned and identified, periods after non-abbreviation words can be designated as sentence boundaries. Then, Punkt performs additional classification to detect abbreviations that are also ends of sentences, ellipses at the ends of sentences, initials, and ordinal numbers. Punkt does not require knowledge of upper and lower case letters, so it is well-suited to languages or data which may not use them.

7.3 Post-normalization

Our final pass, post-normalization, examines the segmented data sentence-by-sentence. First, any sentences in languages which do not use the Latin script but that nonetheless contain words in the Latin alphabet are removed. We also remove sentences containing URLs and put abbreviations into a standard form, using underscores instead of periods. Finally, we replace numerals with their written-out form, where possible, based on the Language Specific Peculiarities document (LSP) provided by Appen Butler Hill to Babel participants.

This type of normalization, while specific to our application, should be reasonable for use in other tasks as well, especially where language modeling is the target.

8 Experiments and Results

Our goal in collecting web data is to supplement language models for ASR and KWS by increasing the lexicon available from the ASR training corpus in order to reduce the number of OOV words available for ASR and KWS. That is, if new words can be added to the lexicon from sources similar in genre to the training and test data, then there is a greater chance that these words can be identified in ASR and KWS on the test corpus. For evaluation purposes here, we calculate OOV reduction by comparing the web-data-augmented lexicon with each of the Babel LLP lexicons for the six Babel OP3 languages – Pashto, Paraguayan Guarani, Igbo, Amharic, Halh Mongolian, and Javanese in Table 1. "LLP" refers to the original lexicon that was distributed with the Limited Language Pack for each language, and "+web" is the union of all of the words in the LLP lexicon and all of the words that we found in the web data. The "%rel.ch" row shows the percent relative change in OOV rate when the web data is added to the lexicon. "OOV KW Rate %" shows the percentage of KWS development queries containing an out-of-vocabulary tokens, both before and after our web data is added to the lexicon. "OOV Hit Rate %" is a similar measure, except that each query term is weighted by the number of times that it actually appears in the development transcripts; in this metric, keywords that appear more often have a greater impact. Finally, "Voc. Size (K)" shows the size of the vocabulary (in thousands of words), before and after adding web data. We see that, for each language, the percentage of OOV queries is significantly reduced; in particular, most Halh Mongolian and Javanese OOV keywords missing from the original lexicons are in fact added to the lexicon by the web data collection.

Language	Lexicon	OOV KW Rate %	OOV Hit Rate %	Voc. Size (K)
Pashto	LLP	24.18	7.35	6.2
	+web	7.44	1.51	2461.6
	%rel.ch	-69.21	-79.39	39693.8
Para-guayan Guarani	LLP	34.84	6.65	9.1
	+web	32.00	5.75	40.3
	%rel.ch	-8.17	-13.66	339.93
Igbo	LLP	30.50	6.52	6.7
	+web	21.74	3.43	50.5
	%rel.ch	-28.71	-47.39	650.1
Amharic	LLP	34.67	9.96	11.6
	+web	32.96	9.27	84.1
	%rel.ch	-4.91	-6.91	627.4
Halh Mongo lian	LLP	32.95	15.67	8.5
	+web	5.37	0.44	2427.6
	%rel.ch	-83.71	-97.16	28450.1
Javanese	LLP	33.61	14.37	5.7
	+web	4.35	0.17	1723.2
	%rel.ch	-87.06	-98.78	30037.3
Dholuo	LLP	31.61	22.26	7.2
	+web	25.46	3.12	48.0
	%rel.ch	-19.45	-85.99	561.6

Table 1: OOV Reduction on Unnormalized Data

While text normalization is important if we are to use the web data for training a language model for ASR, we must also consider the extent to which normalization processes data may in fact remove useful words. Table 2 shows OOV reduction when adding the normalized web data collected. Surprisingly, using the normalized web data to augment the vocabulary actually helps in some instances over using the unnormalized data.

Language	Lexicon	OOV KW Rate %	OOV Hit Rate %	Voc. Size (K)
Pashto	LLP	24.18	7.35	6.2
	+web	5.73	0.75	801.9
	%rel.ch	-76.32	-89.74	12863.6
Paraguayan Guarani	LLP	34.84	6.65	9.2
	+web	31.35	5.64	22.8
	%rel.ch	-10.02	-15.21	149.1
Igbo	LLP	30.50	6.52	6.7
	+web	20.98	3.33	28.1
	%rel.ch	-31.21	-48.91	317.8
Amharic	LLP	34.67	9.96	11.6
	+web	9.54	1.59	646.7
	%rel.ch	-72.48	-83.99	5495.4
Halh Mongolian	LLP	32.95	15.67	8.5
	+web	5.28	0.44	1190.1
	%rel.ch	-83.96	-97.19	13896.8
Javanese	LLP	33.61	14.37	5.7
	+web	4.10	0.15	950.1
	%rel.ch	-87.81	-98.94	16516.7
Dholuo	LLP	31.61	22.26	7.3
	+web	25.22	3.10	24.0
	%rel.ch	-20.23	-86.07	231.1

Table 2: OOV Rate on Normalized Data

This is probably because the removal of special characters and punctuation attached to words results in exact matches for keywords.

Finally, we are interested in seeing the individual contribution of each of the web data genres we collected. Table 3 shows the percent relative reduction in OOVs for both OOV keywords and OOV hit rate in the development data when adding our normalized web data, by language and by genre. It is apparent that the genre that best reduces OOVs varies by language, but tweets were the most generally useful, resulting in the largest OOV reduction for Pashto, Igbo, Halh Mongolian, Javanese, and Dholuo. In fact, tweets were the only useful genre for Dholuo. Paraguayan Guarani saw the largest OOV reduction from forum posts, and Amharic from blogs.

9 Conclusions and Future Research

We have presented a system for collecting conversational web text data for Low Resource Languages. Our system gathers data from a variety of text sources (blogs, forums, Twitter, TED talks) which have proven to be useful for substantially reducing OOV rates for language models based on telephone conversations in a KWS task. Despite the noisy and highly variable nature of text found on the web, by including language identification and text normalization as part of our pipeline, we can be much more confident that the data we collect is likely to be in the target language. Our results have reduced OOV rates for KWS in LRLs significantly, resulting in significantly higher KWS scores. Our future work will explore additional sources for conversational web data, such as Facebook pages and other public social media. We also plan to release our system in the near future as an open source tool for the entire research community.

Language	%rel.ch	Blogs	Forums	TED	Tweets
Pashto	KW	-64.59	-64.48	3.77	-73.20
	Hits	-79.57	-79.57	-8.00	-87.65
Paraguayan Guarani	KW	-4.70	-4.70	n/a	-6.44
	Hits	-8.09	-8.44		-10.39
Igbo	KW	-3.47	-0.42	-0.14	-30.37
	Hits	-9.97	0.29	-0.18	-47.86
Amharic	KW	-66.09	-60.44	-4.30	-61.30
	Hits	-76.42	-72.61	-6.35	-76.12
Halh Mongolian	KW	-73.11	-72.98	-28.16	-82.32
	Hits	-95.16	-95.33	-74.94	-96.86
Javanese	KW	-77.26	-73.12	n/a	-83.17
	Hits	-97.16	-96.42		-97.82
Dholuo	KW	0.0	0.0	n/a	-20.23
	Hits	0.0	0.0		-86.07

Table 3: OOV Rates for Languages by Genre

10 Acknowledgements

This work is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD/ARL) contract number W911NF-12-C-0012. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

References

Nicholas Andrews, Jason Eisner, and Mark Dredze. 2012. Name phylogeny: A generative model of string variation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 344–355. Association for Computational Linguistics.

Marco Baroni and Silvia Bernardini. 2004. Bootcat: Bootstrapping corpora and terms from the web. In *LREC*.

Ivan Bulyko, Mari Ostendorf, and Andreas Stolcke. 2003. Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixtures. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003–short papers-Volume 2*, pages 7–9. Association for Computational Linguistics.

Ivan Bulyko, Mari Ostendorf, Manhung Siu, Tim Ng, Andreas Stolcke, and Özgür Çetin. 2007. Web resources for language modeling in conversational speech recognition. *ACM Transactions on Speech and Language Processing (TSLP)*, 5(1):1.

F. H. Calderon C. Argueta and Y-S. Chen. 2016. ultilingual emotion classifier using unsupervised pattern extraction from microblog data. *JIntelligent Data Analysis*, 29(6).

William B Cavnar, John M Trenkle, et al. 1994. N-gram-based text categorization. *Ann Arbor MI*, 48113(2):161–175.

Mathias Creutz, Sami Virpioja, and Anna Kovaleva. 2009. Web augmentation of language models for continuous speech recognition of sms text messages. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 157–165. Association for Computational Linguistics.

Clément De Groc, Xavier Tannier, and Javier Couto. 2011. Grawltcq: terminology and corpora building by ranking simultaneously terms, queries and documents using graph random walks. In *Proceedings of TextGraphs-6: Graph-based Methods for Natural Language Processing*, pages 37–41. Association for Computational Linguistics.

Jonathan G Fiscus, Jerome Ajot, John S Garofolo, and George Doddington. 2007. Results of the 2006 spoken term detection evaluation. In *Proc. SIGIR*, volume 7, pages 51–57. Citeseer.

Ankur Gandhe, Long Qin, Florian Metze, Alex Rudnicky, Ian Lane, and Matthias Eck. 2013. Using web text to improve keyword spotting in speech. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 428–433. IEEE.

Mary Harper. 2011. Iarpa solicitation iarpa-baa-11-02. *IARPA BAA*.

D. Hoogeveen and G. De Pauw. 2011. corpuscollie-a web corpus mining tool for resource-scarce languages. In *Proceedings of Conference on Human Language Technology for Development. Alexandria, Egypt*, pages 44–49.

Rukmini Iyer, Mari Ostendorf, and Herb Gish. 1997. Using out-of-domain data to improve in-domain language models. *Signal Processing Letters, IEEE*, 4(8):221–223.

Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525.

Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. 2010. Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 441–450. ACM.

Marco Lui and Timothy Baldwin. 2014. Accurate language identification of twitter messages. *EACL 2014*, pages 17–25.

Marco Lui, Ned Letcher, Oliver Adams, Long Duong, Paul Cook, Timothy Baldwin, and NICTA Victoria. 2014. Exploring methods and resources for discriminating similar languages. *COLING 2014*, page 129.

Gideon Mendels, Erica Cooper, Victor Soto, Julia Hirschberg, Mark Gales, Kate Knill, Anton Ragni, and Haipeng Wang. 2015. Improving speech recognition and keyword search for low resource languages using web data. *Proc. Interspeech, Dresden, Germany*.

Tim Ng, Mari Ostendorf, Mei-Yuh Hwang, Man-Hung Siu, Ivan Bulyko, and Xin Lei. 2005. Web-data augmented language models for mandarin conversational speech recognition. In *ICASSP (1)*, pages 589–592.

Jeremy Nicholson and Timothy Baldwin. 2006. Interpretation of compound nominalisations using corpus and web statistics. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, pages 54–61. Association for Computational Linguistics.

Bo Pang and Sujith Ravi. 2012. Revisiting the predictability of language: Response completion in social media. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1489–1499. Association for Computational Linguistics.

Uwe Quasthoff, Matthias Richter, and Christian Biemann. 2006. Corpus portal for search in monolingual corpora. In *Proceedings of the fifth international conference on language resources and evaluation*, volume 17991802.

Ruhi Sarikaya, Agustin Gravano, and Yuqing Gao. 2005. Rapid language model development using external resources for new spoken dialog domains.

Kevin P Scannell. 2007. The crúbadán project: Corpus building for under-resourced languages. In *Building and Exploring Web Corpora: Proceedings of the 3rd Web as Corpus Workshop*, volume 4, pages 5–15.

SIL. 1999. English wordlists. `http://www01.sil.org/linguistics/wordlists/english/`. Accessed: 2015-09-30.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Gohneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, et al. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of The First Workshop on Computational Approaches to Code Switching*, pages 62–72. Citeseer.

Yu Wang, Tom Clark, Eugene Agichtein, and Jeffrey Staton. 2014. Towards tracking political sentiment through microblog data. *ACL 2014*, page 88.

Le Zhang, Damianos Karakos, William Hartmann, Roger Hsiao, Richard Schwartz, and Stavros Tsakalidis. 2015. Enhancing low resource keyword spotting with automatically retrieved web documents. In *Sixteenth Annual Conference of the International Speech Communication Association*.