Improving Speech Recognition and Keyword Search for Low Resource Languages Using Web Data

Gideon Mendels[1], *Erica Cooper*[1], *Victor Soto*[1], *Julia Hirschberg*[1]
Mark Gales[2], *Kate Knill*[2], *Anton Ragni*[2], *Haipeng Wang*[2]

[1]Columbia University, New York, USA
[2]University of Cambridge, Cambridge, UK
gm2597@columbia.edu, {ecooper, vsoto, julia}@cs.columbia.edu
{mjfg, kate.knill, ar527, hw443}@eng.cam.ac.uk

Abstract

We describe the use of text data scraped from the web to augment language models for Automatic Speech Recognition and Keyword Search for Low Resource Languages. We scrape text from multiple genres including blogs, online news, translated TED talks, and subtitles. Using linearly interpolated language models, we find that blogs and movie subtitles are more relevant for language modeling of conversational telephone speech and obtain large reductions in out-of-vocabulary keywords. Furthermore, we show that the web data can improve Term Error Rate Performance by 3.8% absolute and Maximum Term-Weighted Value in Keyword Search by 0.0076-0.1059 absolute points. Much of the gain comes from the reduction of out-of-vocabulary items.

Index Terms: web resources, web scraping, keyword search, low-resource languages

1. Introduction

In large-vocabulary speech recognition systems, the language model (LM) is a key component. Large amounts of text in the target domain are required in order to establish accurate n-gram information. In Low Resource Languages (LRLs), where there may only be a few hours of transcribed audio, transcripts may not be sufficient for building adequate LMs, so that additional data from other sources must be collected to augment these transcripts. The web has become a promising source of found data for training LMs in diverse languages.

We present results of web scraping of data for LRLs in the context of the IARPA Babel program [1]. The task we address is the rapid creation of speech recognition (ASR) and keyword search (KWS) technologies for diverse languages for which only a small amount of transcribed training data are available. In the Babel evaluation task, a query term (word or phrase) is provided in text and all matching utterances in the audio corpus are returned to the user. First, it is important to identify text sources that are genre-appropriate to the speech that the system will encounter. Second, with the informality of much of the data that can be found, steps must be taken to clean and normalize the data. Third, as the web is an international and multilingual community, language that is not in the target language must be identified, to distinguish *code-switched* language from the target. We show that innovative techniques in web scraping can improve the performance of ASR and KWS significantly.

Much work has been carried out to supplement LMs with out-of-domain data, specifically with web data [2], combining out-of-domain data with domain-dependent data to improve statistical LM performance. [3] introduced the use of Bayesian updates for online LM using the web as a source of information. [4] achieved better n-gram coverage from querying the web, estimating n-gram counts from page counts and combined LMs using linear and geometric interpolation, exponential methods, and thresholding. To improve automatic speech recognition (ASR) performance on conversational speech, [5] showed that web data which was filtered to match the style of the domain could provide boosts in recognition, and that class-dependent interpolation of LMs could outperform classical linear interpolation. [6] extended previous work for Chinese and used explicit topic modeling for LMs, while [7] continued work on topic modeling using web data with a mixture of topic-independent modeling and a specific topic model. [8] crawled appropriate texts from RSS feeds and Twitter for LM creation and vocabulary adaptation. Working in the domain of SMS data, [9] described an efficient query selection algorithm for retrieval of web text data for LM augmentation for general and user specific vocabularies. While most work has used perplexity as similarity measure between in-domain data and web data [10], [11] proposed the use of BLEU scores.

To improve keyword search performance, [12] proposed a language-dependent scheme to leverage IPA web resources to derive pronunciation lexica. [13] used web data to improve KWS term-weighted value (TWV) performance. They collected data from three different sources: Wikipedia data for each language, the top 30 Google results from queries seeded with unigrams and bigrams extracted from the training data, and news articles for one of the languages. To filter this data they proposed two different approaches. In the first, each sentence was scored by the weighted difference of the perplexity of the sentence in the base language model and the sum of perplexity values of each Out-of-Vocabulary (OOV) word in the sentence as scored by the web language model. The top scored sentences were then used for subsequent LM and the OOV words were added to the vocabulary. In their second approach the authors used an OOV detection scheme [14] trained on web data to discover OOV tokens in the evaluation data. These newly-discovered OOV words were then matched to potentially similar OOV words in the web data, the goal being to add words to the vocabulary that might be similar to in-domain words. In this case the final language model was trained on all collected data with the expanded vocabulary. The authors reported up to 50% OOV rate reduction and a gain of between 2% and 4% ATWV points.

In our approach, we collect data from a variety of genres which we hypothesized might be relevant to our task, KWS in telephone conversations, testing news, blogs, TED talk transcripts, and movie subtitles. We focus on building tools that can be used to collect data in an automatic and language-independent way by relying on the regular structure of websites hosted on blogspot.com and wordpress.com. We also emphasize precision when collecting our data, by verifying that it belongs to the target language using tools for language detection. We find that the inclusion of our filtered web data achieves improvements in OOV reduction, ASR performance, and keyword search Maximum Term Weighted Value (MTWV).

2. Data

Our research makes use of the IARPA Babel Program language collection IARPA-babel{205b-v1.0a Kurmanji Kurdish, 207b-v1.0b Tok Pisin, 302b-v1.0a Kazakh, 303b-v1.0a Telugu, 304b-v1.0b Lithuanian} very limited language packs (VLLP). We focus on the conversational speech for each language, telephone speech of about 10 minutes in length between two speakers recorded on separate channels, in a variety of recording conditions. The speakers are diverse in terms of age and dialect and the gender ratio is approximately even. There are 80 hours of audio for each language, of which 3 hours are transcribed. We have collected text data from a variety of sources, such as blogs, online news, and TED (Technology, Education, Design) talk transcripts[1] translated from English into the target languages through the Open Translation Project. We have also used subtitle data gathered from the Open Subtitles Database[2] [15] downloaded by BBN and shared with the Babel participants. Table 1 shows the token counts for each language.

3. Webscraping

To scrape large amounts of conversational data several approaches were implemented. While these approaches vary in details, they each involve five main steps: 1) Finding an appropriate source in terms of genre; 2) Designing a crawler based on the individual URL structure of the source; 3) Extracting only relevant textual data, stripping HTML tags and other meta-information; 4) Removing data not in the target language using a trained language identifier; and 5) Saving the data in plain text while maintaining a log of the exact source.

We began with a manual approach in which much of the work involved finding sources with a large amount of data in the target language and preparing a custom crawler and scraper for each source. Some of the sources were multilingual, such as TED talks, and others were single language, such as blogs and news sources. Once a source was found, we examined the URL structure and designed a crawler that would be able to retrieve all the pages/posts on that site. The Document Object Model (DOM) structure of each source was examined to find the elements that contain the textual data. Using Jsoup [16], a Java HTML Parser, and CSS selectors, we retrieved the raw data. Then this data was fed into Google's Compact Language Identifier (CLD) [17], a pre-trained naive Bayes classifier that allowed us to reject data not in the target language.

Since these steps were quite labor-intensive, we experimented with several approaches to optimizing the process. One approach was to use Google Search queries to target blogs from blogspot.com and wordpress.com. All the blogs hosted on those platforms have an opt-out public RSS feed that allowed us to use the same scraper logic for all blogs. We also used the language parameters provided by Google Search to filter out irrelevant languages. A typical query to retrieve a list of blogs in Lithuanian from blogspot.com would be:

`https://www.google.com/search?as_q=`
`&lr=lang_lt&as_sitesearch=blogspot.com`

where the `lr` parameter defines the language code. The extracted data was examined by CLD to verify the language again, because, often, even if a blog was identified by a particular language code, it nevertheless contained data in other languages. This method allowed us to scrape thousands of blogs fairly quickly. We used RSS GUID as a unique identifier to make sure we were not saving the same post twice (since Google is likely to return more than one result from each blog). This approach was successful, but only viable for the languages Google search can filter by, which in our case was only Lithuanian.

In languages Google search could not filter, a slightly different approach was used to obtain results. Instead of using Google's language filter, we seeded the search with a word in the target language. Although the results were not filtered by language, the query reduced the search results to mostly results in the target language. By seeding each search with one word out of the 1000 most common words in the language, and limiting the scope to pages only from blogspot.com, we were able to retrieve more blogs. The results were later filtered by CLD.

4. Preprocessing and Normalization

4.1. Text Normalization

Our process for cleaning and normalizing the text collected from the web involved three steps: 1) pre-normalization, a first pass in which non-standard punctuation was standardized; 2) tokenization, which was accomplished with the Punkt module of NLTK [18]; and 3) post-normalization, in which sentence-by-sentence cleaning of any remaining out-of-language text and standardization of numerals and abbreviations was done.

During the pre-normalization phase, we first removed list entries and titles, since those generally are not full sentences. We replaced non-standard characters with a standard version - this includes ellipses, whitespace, hyphens, and apostrophes. Hyphens and apostrophes were removed as extraneous punctuation, except when they occurred word-internally, as part of a hyphenated word or a contraction. Finally, any characters that are not part of the language's character set, the Latin character set, numerals, or allowed punctuation were removed. This removes special characters such as symbols. Latin characters were preserved, even for languages that use a different alphabet, in order to enable the more accurate removal of entire sentences containing foreign words in a later stage of normalization.

Next, we performed tokenization using the Punkt module of NLTK. Punkt uses a language-independent, unsupervised approach to sentence boundary detection. It works by first learning which words are abbreviations as opposed to sentence final words. It uses three criteria to characterize and identify abbreviations: 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, periods after words that are not identified as abbreviations can be designated as sentence boundaries. Then,

[1] `http://www.ted.com/translate/about`
[2] `http://www.opensubtitles.org`

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 a language that does not use them, such as Telugu.

For our sentence tokenization, we used Punkt to learn a segmentation over all of the data for each genre of each language.

Our second pass, post-normalization, looked at the newly-segmented data sentence-by-sentence. First, any Telugu or Kazakh sentences containing words using the Latin alphabet were removed. We also removed any sentence which contains a URL, and we transformed abbreviations into a standard form, using underscores instead of periods. Finally, we replaced numerals with their written-out form, where possible and appropriate, based on the Table of Numbers included with the Language Specific Peculiarities document (LSP) for each language, provided by Appen Butler Hill.[3]

4.2. Language Filtering

Due to the nature of web data, our results were highly dependent on language identification and filtering. We used a combination of CLD [17] and our own language identifier, created from LingPipe [19]. Since CLD does not support Kurmanji or Tok Pisin, we hadwere forced to implement our own classifier. The classifier was constructed based on LingPipe's NGramProcess which constructs a dynamic classifier over the specified categories, using process character n-gram models of the specified order. We trained the model on 101 languages, including the six languages of interest. The training data for the other languages was retrieved from the Leipzig corpora [20]. Where available, we chose the web data to match the genre; if not available, the news crawl data was used. While in most cases the general accuracy of the classifier is important, we used cross-validation to optimize the reduction of the number of false positives and false negatives for our target languages. Our model was trained on a subset of 300,000 characters from each language with a 5-gram count, and gave us an overall accuracy of over 96.9%.

5. System Specification

For ASR, language dependent discriminatively trained Tandem SAT systems were used [21]. Training and recognition were performed using HTK V3.4.1 [22]. Graphemic models were created since no phonetic lexicons were provided for the Babel VLLPs. The graphemes, and associated attributes, were automatically determined from the unicode text as described in [23]. State-root position dependent decision trees were used to tie states enabling rare and unseen graphemes to be modelled. The 107-D Tandem features consisted of: 62-D bottleneck features generated from a MRASTA based multilingual DNN, which was initially trained with the data from 11 Babel BP and OP1 full language packs and then fine-tuned on the target language [24, 25]; 39-D HLDA transformed 52-D PLP+Δ+Δ^2+Δ^3; pitch+Δ+Δ^2 and probability of voicing (POV)+Δ+Δ^2. Pitch and POV were estimated using the Kaldi toolkit [26].

Word based bigram language models (LM) were used in decoding, with trigram LMs used for lattice rescoring and confusion network generation. Each LM was trained with modified Kneser-Ney smoothing using the SRI LM toolkit [27]. The VLLP LM word list corresponded to the VLLP training transcriptions word list. For the larger web data collections, frequency based cut-offs were used to restrict the size of the word list. Generally, the top (approximately) 100,000 words were selected by count. Graphemic lexicons were produced by automatically mapping the word lists to the corresponding graphemic set. Two decoding passes were run, with a first pass decoding with a speaker independent MPE trained Tandem system being used to generate the supervisions for the speaker transforms for the Tandem SAT system.

Weighted finite state transducer based indexing and search were used for keyword spotting (KWS) with a KWS system provided by IBM [28, 29]. Search was performed on the bigram-decoded lattices. Word-level search was performed for IV terms, followed by grapheme level cascade search on the IV terms for which no hits were returned and a grapheme level search for OOV terms. The LM scores were ignored in the grapheme searches. To boost the OOV detection performance, query expansion using a full grapheme-to-grapheme confusion matrix [30] was applied (NBestP2P). NBestP2P was set to 100 for all experiments in this paper. The KWS search returned approximate posterior probabilities of each search term occurring at a particular point in time. Before MTWV scoring these values were further normalised using a sum-to-one approach to ensure that the sum over the test set of the scores for each keyword sum to unity. More details of the approach are given in [30].

6. Experiments & Results

To evaluate the usefulness of the web data in different genres on the language modeling task, we trained interpolated language models on the VLLP training partition and each genre subset, and tested it on the VLLP tuning set. Table 1 shows the interpolation weights for each language model and for each language. For every language, VLLP has the largest weight, since it matches its own style. For the rest of the genres, it can be observed that, when available, *Blog* data usually has the largest weights, which we attribute to personal blog entries being close in style to conversational speech. *Subtitles* are usually highly weighted also. One could argue that these are a better match for conversational speech than *Blogs*, and thus should obtain larger weights. However, *Blog* data is always significantly more abundant (table 1, last column) than *Subtitles* data. *TED* talks transcripts prove to be a significant addition for Lithuanian, due to the significant size of the data available for that language, but not for Telugu and Kazakh. Similarly, *News* data does not help to fit the VLLP tuning set, except for Kurmanji, where it is the only web data available. It can be concluded that, in the presence of data similar in style to conversational speech (*Blogs, Subtitles*), the rest of the genres do not add significant gains to the LM performance.

Besides the LM performance, the addition of web data has important effects on the ASR vocabulary and the IV/OOV decoding performance of the KWS system. Table 2 shows the fraction of OOV keywords in the keyword list, the hit rate of the OOV keywords in the tuning set, and the vocabulary size for both the VLLP dataset and the VLLP with the added web data. For each language, the percentage of OOV queries (column 3) in the keyword list is significantly reduced; Kazakh specifically obtains a relative reduction of OOV queries of 72%, and Lithuanian close to 64%, out of about four thousand development queries for each language. Even the language with the lowest relative reduction (Telugu), sees its OOV queries reduced by 37.31%. The hit rate of OOV queries on the tuning set (column 4) sees similar relative reductions, although, in general, the absolute numbers are lower due to the OOV queries being less

[3] http://www.appenbutlerhill.com

Language	Language Sources	LM Weights	Number Tokens (K)
Kurmanji	VLLP	0.970	82.1
	News	0.030	1617.7
Tok Pisin	VLLP	0.984	78.7
	Blogs	0.016	483.0
Kazakh	VLLP	0.845	61.4
	Blogs	0.100	708.4
	Subtitles	0.055	9.6
	TED	0.000	25.7
Telugu	VLLP	0.838	57.0
	Blogs	0.128	2625.6
	News	0.000	893.0
	Subtitles	0.024	24.9
	TED	0.000	18.8
Lithuanian	VLLP	0.887	73.7
	Subtitles	0.067	805.0
	TED	0.045	551.7

Table 1: *a) Interpolation Weights, all bigrams (LM for above numbers), tuned in VLLP tuning set b) Number of tokens in each genre web dataset.*

frequent in the dataset.

Furthermore, with the addition of web data, the decoding vocabulary size is relatively increased by 921-2286%, depending on the language.

Language	LM	OOV KW Rate %	OOV Hit Rate %	Voc. Size (K)
Kurmanji	VLLP	29.19	9.59	3.6
	+Web	14.84	5.34	85.9
	%Rel.Ch	-49.16	-44.32	+2286
Tok Pisin	VLLP	23.85	3.88	1.9
	+Web	13.66	2.33	19.4
	%Rel.Ch	-42.73	-39.95	+921
Kazakh	VLLP	35.33	13.43	5.3
	+Web	9.88	4.48	113.3
	%Rel.Ch	-72.04	-66.64	+2038
Telugu	VLLP	43.6	23.75	7.1
	+Web	27.33	14.49	87.7
	%Rel.Ch	-37.31	-38.99	+1135
Lithuanian	VLLP	41.57	19.20	5.4
	+Web	15.00	9.25	111.2
	%Rel.Ch	-63.92	-51.83	+1959

Table 2: *a) Rate of OOV Keywords/Phrases on the KW list, b) OOV Rates on VLLP tuning set, c) Language Model Vocabulary Sizes*

With respect to the ASR performance, no change in Term Error Rate (TER) is noted on Tok Pisin, while modest improvements are obtained on Kurmanji, Kazakh and Telugu with reductions of 0.1, 0.7 and 0.8 absolute points of TER. More impressive is the absolute reduction of 3.8 points for Lithuanian. Even more significant are the improvements obtained for keyword search. The MTWV performance on Kazakh, Telugu and Lithuanian was increased by 0.0489, 0.0459 and 0.1059 absolute points. Although there were gains in IV MTWV performance too, most improvements come from the OOV queries. Column 5 shows that while Kurmanji and Tok Pisin increased their OOV MTWV performance by 0.024 points, the rest of the languages saw increments of 0.1255, 0.1 and 0.2342 points.

Language	Language Model	TER (%)	MTWV iv	MTWV oov	MTWV tot
Kurmanji	VLLP	75.8	0.1705	0.0957	0.1488
	+ Web	75.7	0.1711	0.1197	0.1564
Tok Pisin	VLLP	55.5	0.3381	0.0909	0.2802
	+ Web	55.5	0.3392	0.1145	0.2866
Kazakh	VLLP	70.9	0.2831	0.0901	0.2152
	+ Web	70.2	0.2901	0.2156	0.2641
Telugu‡	VLLP	83.4	0.2110	0.0197	0.1278
	+ Web	82.6	0.2153	0.1197	0.1737
Lithuanian	VLLP	69.0	0.4036	0.1672	0.3047
	+ Web	65.2	0.4160	0.4015	0.4106

Table 3: *Performance of graphemic Tandem-SAT († Tandem-SAT-GAUSS, ‡ Tandem-SAT-PI) VLLP acoustic models, Aachen (ML11) features. KWS IV/OOV split from VLLP LM.*

7. Conclusions & Future Work

We have presented results on the impact of adding genre-appropriate, filtered text web data for the task of keyword search on conversational speech for low-resource languages. We scrape a variety of text sources (news, blogs, TED talks transcripts, movie subtitles) and build interpolated language models using this data. Compared to a baseline system, the new language models reduce the hit rate of OOV keywords on the tuning set by 39-66% relative points, depending on the language. Using an Automatic Speech Recognition system with a graphemic acoustic model, the inclusion of the web data helps improve the TER performance up to 3.8 absolute points on Lithuanian. The MTWV measure is further improved for every language by 0.0076-0.1059 absolute points, with the most gain coming from the OOV keywords (between 0.0236 and 0.2343 absolute points for Tok Pisin and Lithuanian respectively).

In future work, we will explore additional sources of conversational web data, such as Twitter and other social media. We are also investigating ways to overcome the limited language filtering options provided by Google Search using filters defined in LingPipe. We have also designed a crawler to index the blogspot.com blog chain with the language ID of each blog by using the "Next Blog" links, which links to another blog to generate a list of all the blogs; and by running language detection on text from each blog, we can annotate that list with a hypothesized language ID. This will provide us with a useful starting point from which to collect data from a new language by querying the database with the language code.

8. 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.

9. References

[1] M. Harper, "IARPA Solicitation IARPA-BAA-11-02," 2011.

[2] R. Iyer, M. Ostendorf, and H. Gish, "Using out-of-domain data to improve in-domain language models," in *IEEE Signal Processing Letters*, no. 8, 1997, pp. 221–223.

[3] A. Berger and R. Miller, "Just-in-time language modelling," in *Proceedings of ICASSP*, 1998, pp. 705–708.

[4] X. Zhu and R. Rosenfeld, "Improving trigram language modeling with the world wide web," in *Proceedings of ICASSP*, 2001, pp. 533–536.

[5] I. Bulyko and M. Ostendorf, "Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixtures," in *Proceedings of HLT-NAACL*, 2003, pp. 7–9.

[6] T. Ng, M. Ostendorf, M. Hwang, M. Siu, I. Bulyko, and L. Xin, "Web-data augmented language models for mandarin conversational speech recognition," in *Proceedings of ICASSP*, 2005, pp. 589–592.

[7] A. Sethy, P. Georgiou, and S. Narayanan, "Building topic specific language models from web data using competitive models," in *Proceedings of Interspeech*, 2005, pp. 1293–1296.

[8] T. Schlippe, L. Gren, N. T. Vu, and T. Schultz, "Unsupervised language model adaptation for automatic speech recognition of broadcast news using web 2.0." in *Proceedings of Interspeech*. ISCA, 2013, pp. 2698–2702.

[9] M. Creutz, S. Virpioja, and A. Kovaleva, "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 ACL (EACL 2009)*. Athens, Greece: Association for Computational Linguistics, March 2009, pp. 157–165.

[10] I. Bulyko, M. Ostendorf, M. Siu, T. Ng, A. Stolcke, and O. Çetin, "Web resources for language modeling in conversational speech recognition," *ACM Trans. Speech Lang. Process.*, vol. 5, no. 1, pp. 1:1–1:25, Dec. 2007.

[11] R. Sarikaya, A. Gravano, and Y. Gao, "Rapid language model development using external resources for new spoken dialog domains," in *Proceedings of ICASSP*, vol. 1, 2005, pp. 573–576.

[12] D. Can, E. Cooper, A. Ghoshal, M. Jansche, S. Khudanpur, B. Ramabhadran, M. Riley, M. Saraclar, A. Sethy, M. Ulinski, and C. White, "Web derived pronunciations for spoken term detection," in *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '09. ACM, 2009, pp. 83–90.

[13] A. Gandhe, L. Qin, F. Metze, A. Rudnicky, I. Lane, and M. Eck, "Using web text to improve keyword spotting in speech," in *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU),*, Dec 2013, pp. 428–433.

[14] L. Qin and A. I. Rudnicky, "OOV word detection using hybrid models with mixed types of fragments," in *Proceedings of INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, 2012.

[15] J. Tiedemann, "Parallel data, tools and interfaces in opus," in *Proceedings of LREC*, no. 4, 2012, pp. 2214–2218.

[16] jsoup: Java html parser. [Online]. Available: http://jsoup.org/

[17] Compact language detector 2. [Online]. Available: https://code.google.com/p/cld2/

[18] T. Kiss and J. Strunk, "Unsupervised multilingual sentence boundary detection," in *Computational Linguistics*, no. 4, 2006, pp. 485–525.

[19] B. Carpenter. Lingpipe dynamiclmclassifier. [Online]. Available: http://alias-i.com/lingpipe/docs/api/com/aliasi/classify/DynamicLMClassifier.html

[20] U. M. R. C. B. Quasthoff, "Corpus portal for search in monolingual corpora," *Proceedings of LREC*, pp. 1799–1802, 2006.

[21] S. Rath, K. Knill, A. Ragni, and M. Gales, "Combining Tandem and Hybrid Systems for Improved Speech Recognition and Keyword Spotting for Low Resource Languages," in *Proceedings of ICASSP*, 2014.

[22] S. J. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK version 3.4.1)*. Cambridge University, 2009, `http://htk.eng.cam.ac.uk`.

[23] M. Gales, K. Knill, and A. Ragni, "Unicode-based graphemic systems for limited resource languages," in *Proceedings of ICASSP*, 2015.

[24] Z. Tüske, J. Pinto, D. Willett, and R. Schlüter, "Investigation on cross- and multilingual MLP features under matched and mismatched acoustical conditions," in *Proceedings of ICASSP*, 2013.

[25] Z. Tüske, D. Nolden, R. Schlüter, and H. Ney, "Multilingual MRASTA features for low-resource keyword search and speech recognition systems," in *Proceedings of ICASSP*, 2014.

[26] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 2011.

[27] A. Stolcke, "SRILM - An Extensible Language Modeling Toolkit," in *Proceedings of ICSLP*, 2002.

[28] B. Kingsbury, J. Cui, X. Cui, M. Gales, K. Knill, J. Mamou, L. Mangu, D. Nolden, M. Picheny, B. Ramabhadran, R. Schüter, A. Sethy, and P. Woodland, "A high-performance Cantonese keyword search system," in *Proceedings of ICASSP*, 2013.

[29] M. Mohri, C. Allauzen, and M. Saraclar, "General indexation of weighted automata – applicaiton to spoken utterance retrieval," in *Proceedings of HLT-NAACL*, 2014, pp. 2345–2349.

[30] L. Mangu, H. Soltau, H.-K. Kuo, B. Kingsbury, and G. Saon, "Exploiting diversity for spoken term detection," in *Proceedings of ICASSP*, 2013.