EFFICIENT SPOKEN TERM DETECTION USING CONFUSION NETWORKS

Lidia Mangu, Brian Kingsbury, Hagen Soltau, Hong-Kwang Kuo and Michael Picheny

IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA

ABSTRACT

In this paper, we present a fast, vocabulary independent algorithm for spoken term detection (STD) that demonstrates a word-based index is sufficient to achieve good performance for both in-vocabulary (IV) and out-of-vocabulary (OOV) terms. Previous approaches have required that a separate index be built at the sub-word level and then expanded to allow for matching OOV terms. Such a process, while accurate, is expensive in both time and memory. In the proposed architecture, a word-level confusion network (CN) based index is used for both IV and OOV search. This is implemented using a flexible WFST framework. Comparisons on 3 Babel languages (Tagalog, Pashto and Turkish) show that CN-based indexing results in better performance compared with the lattice approach while being orders of magnitude faster and having a much smaller footprint.

Index Terms— keyword search, spoken term detection, keyword spotting, audio indexing, confusion networks

1. INTRODUCTION

One of the fundamental problems in automatic speech processing is finding a spoken or written term in a collection of audio recordings. Given the vast amount of existing spoken information, with more being produced every day, there is an increasing need for small indices and fast search.

Typically, state-of-the-art spoken term detection (STD) systems work in two phases (1) transforming the speech into text format using an automatic speech recognition system (ASR), and (2) building an index from the text. The simplest textual format is the 1-best hypothesis from the ASR system. This approach will result in good STD performance if the speech recognition system has low word error rate. But most state-of-the-art STD systems benefit from having a richer ASR output representation. Several retrieval methods dealing with multiple hypotheses from an ASR system have been proposed, with lattices and confusion networks [1] being frequently used for building STD indices [2, 3, 4, 5, 6, 7, 8, 9, 10]. The drawback of this approach lies in its inability to find terms that are not in the dictionary of the speech recognizer. Approaches based on sub-word units (phone, graphone, syllable, morph) are widely used to solve the OOV issue. Retrieval consists of searching for the sequence of sub-words representing the OOV term in a sub-word index. Popular approaches are based on search in sub-word decoding output [6, 11, 12] or search on the sub-word representation of the word decoding [13, 5]. To compensate for the errors made by the ASR system, the query term can be expanded using a sub-word confusability model [13, 14]. Since subword-based indices generally yield a lower precision for IV queries compared with word-based ones, the word and subword indices are either used separately for IV and OOV search, respectively [13, 5], or combined into one index [7, 15].

In this work we describe a Weighted Finite State Transducer (WFST) STD architecture in which a word index created from CNs is sufficient for high-performance IV and OOV retrieval. By replacing lattices with confusion networks which are much smaller, and eliminating the need for sub-word units in the index, we ensure a very small footprint index. The organization of this paper is as follows: Section 2 and 3 describe the indexing and search in the proposed architecture. An overview of the task, metric, and ASR system used for indexing is given in Section 4. Section 5 shows our experiments and results and we conclude in Section 6.

2. CONFUSION NETWORK BASED INDEXING

In this section we describe the CN-based WSFT word index which will be used for both IV and OOV keyword search. CNs have a linear structure, representing the competing word hypotheses and their posterior probabilities in consecutive time intervals (*confusion bins*).

A word index containing all the information needed for keyword search (audio file identity, start time, end time, and word label) is constructed from confusion networks using the following steps.

1. Each CN produced by an ASR system is compiled into a weighted finite state transducer (CN FST) having the same topology, input labels that are the words on each arc in the CN, output labels that encode the start time ($Tstart$) and end time ($Tend$) of each arc as $Tstart - Tend$ strings, and costs that are negative log CN posteriors for each arc. Some bins in a CN have deletion/epsilon (*eps*) arcs. Silence, hesitations and other filler words are not written into the index; instead, they contribute to the posterior probability of these epsilon arcs. I.e., the score of a deletion in a bin will be 1 minus the sum of posteriors of *real* words, and any skip over a CN bin will be penalized according to this.

2. In order to be able to access any substring of words in the CN FST i produced in the previous step, we add a new start node, S_i, with zero-cost epsilon-arcs connecting S_i to each node in i, and a new end node, E_i, with zero-cost epsilon-arcs connecting each node in i to E_i.

3. The final single index is obtained by creating a new start node, S, that is connected to each S_i by zero-cost arcs with input label epsilon and output label i (or audio file id), and a new end node, E, that is connected to each E_i by zero-cost epsilon-arcs.

Figure 1 shows the CN-based index. Having this structure, one can retrieve any word or sequence of words from the original CNs, and the corresponding time interval and audio file id. This approach is similar to the indexing approach described in [13], in which the focus was lattice indexing.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

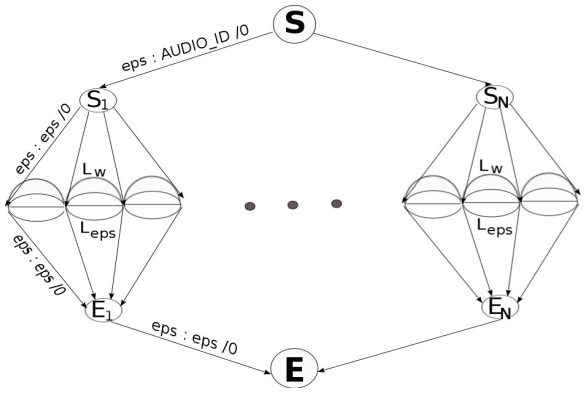

Fig. 1. CN-WFST index.

3. SEARCH

3.1. In-Vocabulary search

Given the linear structure of the CNs, searching for an in-vocabulary keyword is straightforward, similar to searching in the 1-best hypothesis of the recognizer. The only difference comes from multi-word queries, which will be found in consecutive positions in the case of the 1-best hypothesis, while they could be found in non-consecutive bins in a CN. In [16] it is mentioned that CNs might not be appropriate for multi-word query search due to the presence of epsilon links. A WFST framework deals with these issues in an elegant way; when multi-word queries are found in non-consecutive bins, their score is decreased according to the probability of the traversed epsilon links. Thus, the epsilon arcs in the index WFST control which bins can be skipped in a confusion network and with what penalty. Also, the semiring chosen for building the index will specify the method for combining the scores of the word components for a multi-word query. If we choose a Log semiring then the scores will be added (i.e. posteriors will be multiplied), or if we choose a Min-Max semiring, the minimum score will be chosen as the score for the entire keyword. The IV search consists in the following steps:

1. The query is converted into a word automaton.
2. The query automaton is composed with the index transducer.
3. Due to the epsilon arcs in the index, the composition will produce sometimes multiple overlapping hits for a query; among those we keep only the one with the highest score.

The output labels in the resulting FST contain everything that is needed to locate the hit: the audio file id and the start/end time.

3.2. Out-of-vocabulary search

The OOV search is very similar to the IV one: the only extra step needed is the conversion of OOV keywords into in-vocabulary words which sound similar. For this extra step we need three transducers: (1) a word to phone transducer (W2P) which is created using a letter to sound model for OOV words and the ASR lexicon for the IV words (for the case in which the multi-word OOV has IV components), (2) a phone confusability transducer (P2P) which specifies pairs of confusable phones and the probability of the confusion, and (3) a phone to word transducer (P2W) built using the ASR lexicon.

After creating these transducers, the OOV search consists of the following steps:

1. Compose the automaton corresponding to the OOV word query with W2P, converting it into a phone automaton P
2. Compose P with P2P, creating an FST which contains all the alternate phone sequences according to the confusion model
3. Extract N-best paths, thus keeping only the most likely phone sequences
4. Compose the result with P2W

An alternate architecture can be obtained by swapping the last two steps, although, it was found to have worse performance.

The result of the last step is a set of in-vocabulary word sequences that can be searched for in the word index as a proxy for the OOV keyword which has no chance to be found. Note that if we use the identity P2P, the final FST contains the decompositions of the OOV word into sequences of IV words, if they exist. For example, if *meanwhile* is the OOV word, and if *mean* and *while* are in vocabulary, we would search for *mean while*. Figure 2 shows an example for the type of word sequences we are searching for in place of the OOV word *Iraqi*. Some of the sequences will be searched for with no penalty, due to the fact that they share a baseform with the OOV word.

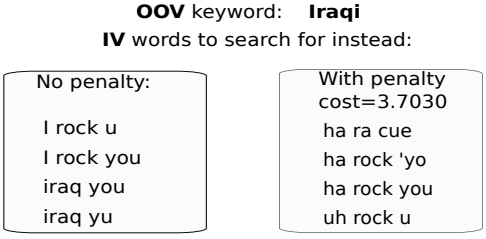

Fig. 2. OOV to IV mapping example for Tagalog

In prior work, a multi-word query is considered OOV if at least one word component is not in the vocabulary. In this situation the query expansion module will expand all the words in the query regardless of their IV/OOV status. In our OOV processing module, we could slightly change the transducers involved in processing the OOV query such that only the OOV query components are expanded, while the IV words are kept the same. The new W2P will contain an identity mapping instead of word-to-phone expansion for IV words. In the P2P and P2W transducers we add a word identity mapping. This procedure has multiple benefits: (1) for a fixed N-best value, we get many more hypotheses for the actual OOV words, due to the fact that we eliminate the confusions for the IV words, and (2) for most tasks, applying a confusability model for the IV words results in loss of precision.

Figure 3 shows the proposed system architecture (CN-STD).

3.3. Score Normalization

As shown in [17], the posting list scores have to be normalized in order to improve performance. For this work we use the same normalization as in [17], with the following slight modification. It is known that high word posteriors in a confusion network are very strong indicators that the word is correct [18]. Therefore we change the normalization such that all the words with a posterior probability above a certain threshold keep their original unnormalized score.

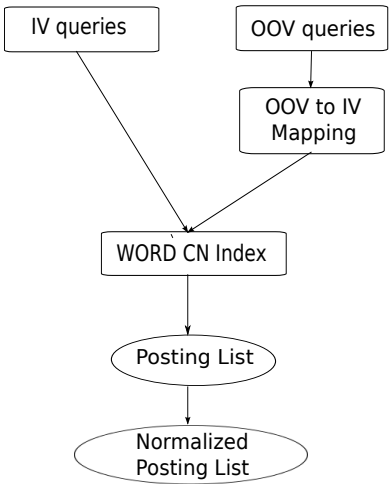

Fig. 3. The architecture of the CN based STD

4. DATA AND ASR SYSTEM DESCRIPTION

We conducted our experiments in the context of the IARPA Babel program [19], which focuses on spoken term detection for low-resource languages. We chose the limited language pack track (LP) of the program, in which only 20 hours of audio, (10 hours of transcribed data) is used for building ASR models and lexicons, making it more interesting for OOV keyword search. In this paper, we focus on 3 of the languages used under this program, namely, Pashto (Pashto LP), Tagalog (Tagalog LP) and Turkish (Turkish LP). For comparison, we also show results for the full language pack track for one language, Tagalog (Tagalog FP) in which training data consists of 100 hours of speech. In preparation for the Babel evaluation we received a dev set (DEV) for training, 20 hours of audio for each language. After the evaluation NIST released references for a portion of the evaluation data (10 hours), which we used together with all the evaluation queries to create another set (EVAL). There are many more queries in EVAL (1700-2100) compared to DEV (300-600). We report results on both DEV and EVAL.

The metric used for the Babel program is Term-Weighted Value (TWV), which was also used to evaluate systems in the NIST 2006 STD Evaluation [20]. We report keyword search performance in terms of maximum Term-Weighted Value (MTWV) which is the best TWV for all values of the decision threshold.

The acoustic model used in these experiments is the IBM Speaker-Adapted DNN (SA DNN) system which uses a deep neural network (DNN) acoustic model with IBM's standard front-end pipeline [21]. The DNN takes 9 frames of 40-dimensional speaker adapted discriminative features as input, contains 5 hidden layers with 1,024 logistic units per layer, and has a final softmax output with 1,000 targets. Training occurs in three phases: (1) layer-wise discriminative pre-training using the cross-entropy criterion, (2) stochastic gradient descent training using back-propagation and the cross-entropy criterion, and (3) distributed Hessian-free training using the state-level minimum Bayes risk criterion [22]. The lexicon was provided with the training data, and the vocabulary contains only words from this data. The language model (LM) is a trigram LM with modified Kneser-Ney smoothing trained only on the acoustic transcripts. The lattices are produced using a dynamic decoder [23], and confusion networks are generated from these lattices. Compared to the original CN generation algorithm [1], we used a faster version with the following differences: (1) the slow total order computation is replaced with a fast local order constraint, (2) time overlap of the clusters to be merged is enforced, and (3) low posterior links are not allowed to participate in the intra-word merging step. The new algorithm is 2-5 times faster than the original, and more robust when the pruning threshold is very low (which is important for STD tasks). Table 1 shows 1.3-1.5% absolute improvements in WER due to the lattice to CN conversion.

WER	Tagalog FP	Tagalog LP	Turkish LP	Pashto LP
Lattice 1-best	53.7	63.9	65.0	65.3
CN 1-best	52.2	62.7	63.7	63.5

Table 1. WER comparison for lattice and CN 1-best.

For simplicity we present results for this acoustic model only, given that it had the best STD performance among all the systems built at IBM during the evaluation, though similar improvements are obtained for a Gaussian-mixture model (GMM) or a speaker independent DNN model.

5. EXPERIMENTS AND RESULTS

The OpenFST Library [24] is used for both indexing and search. Among the semirings we compared for the task, we chose the Log semiring which performed the best. Regarding the phone confusability transducer, there are many methods for creating it [17, 25, 14]. The evaluation system used a simple method with the following steps: (1) Create Viterbi alignments of the training data transcripts using an acoustic model, (2) Decode the training data using the same acoustic model and a unigram LM, and (3) Compute state-level confusability by comparing the two sets of alignments from the ground truth and decoding hypotheses, respectively. This is converted to phone-level confusability. As a baseline for the CN based STD we use a state-of-the-art lattice WFST STD architecture which we successfully deployed in both DARPA RATS and IARPA Babel evaluations [13, 17, 25, 26]. In this architecture a word index built from lattices is used for IV search and a phone index is used for OOV search, after the OOV queries are expanded using the phone confusability transducer. We use the same phone confusability transducer for both lattice and CN approach. The number of N-best phone sequences to be retained for each OOV word is optimized separately for each framework. We compare with a strong baseline, these are the systems that we submitted in the Babel evaluation. Our results for Pashto (Table 2), Turkish (Table 3) and Tagalog (Table 4), show that CN-STD performs the same or better for both IV and OOV terms. This conclusion holds also for the full pack condition in which the vocabulary is 3.5 times larger and the WER is 10% absolute better (Table 5). The new approach leads to up to 12% relative MTWV improvement.

System	DEV			EVAL		
	IV	OOV	ALL	IV	OOV	ALL
Lattice-STD	0.2085	-0.0351	0.1846	0.2379	0.0481	0.2122
CN-STD	0.2312	0.0044	0.2107	0.2464	0.0567	0.2208

Table 2. MTWV comparison of the lattice and CN STD system on Pashto LP.

System	DEV			EVAL		
	IV	OOV	ALL	IV	OOV	ALL
Lattice-STD	0.4450	0.0591	0.3424	0.3320	0.0419	0.2610
CN-STD	0.4460	0.1001	0.3526	0.3331	0.0589	0.2646

Table 3. MTWV comparison of the lattice and CN STD system on Turkish LP.

System	DEV			EVAL		
	IV	OOV	ALL	IV	OOV	ALL
Lattice-STD	0.2868	0.1601	0.2586	0.3441	0.0796	0.2511
CN-STD	0.2945	0.1601	0.2639	0.3452	0.0799	0.2512

Table 4. MTWV comparison of the lattice and CN STD system on Tagalog LP.

System	DEV			EVAL		
	IV	OOV	ALL	IV	OOV	ALL
Lattice-STD	0.5281	0.1636	0.5021	0.5673	0.1079	0.5273
CN-STD	0.5426	0.2880	0.5249	0.5718	0.1307	0.5330

Table 5. MTWV comparison of the lattice and CN STD system on Tagalog FP.

Regarding the speed and size of the proposed architecture, Table 6 shows that the CN-STD is orders of magnitude smaller and faster than the lattice STD. Note that the search time difference is much larger for the LP track which has 4 times more OOVs, and therefore many more FST compositions with the large phone-level index.

	System	Indexing Time	Search Time	Index Size
Tagalog LP	Lattice	576 mins	7233 mins	4264 Mb
	CN	8 mins	360 mins	201 Mb
Tagalog FP	Lattice	548 mins	468 mins	3346 Mb
	CN	5 mins	60 mins	157 Mb

Table 6. Running time and footprint comparison for a DEV+EVAL run (3963 queries searched in 30 hours of audio) for Tagalog.

For both IV and OOV posting lists we can eliminate the hits with scores below a certain threshold (for example 1e-07 for IV and 1e-08 for OOV) before normalization. This thresholding not only reduces the size of the final posting list, but occasionally results in small improvements in performance. Alternate methods for computing the scores of the multi-word query hits are proposed in [15, 8]. In [15] the posterior probability of a word is multiplied by the inverse of the rank r of the word in a confusion bin. [8] reports that $1/r$ by itself works as well if not better. We compared these alternate strategies on the Tagalog FP task. Table 7 shows that the rank by itself works surprisingly well, but none of these alternate strategies work better than using the posterior probability by itself. It could be that other combinations of the posterior probability and rank work better, this is something to explore in the future.

System	DEV			EVAL		
	IV	OOV	ALL	IV	OOV	ALL
$Posterior$	0.5426	0.2880	0.5249	0.5718	0.1307	0.5330
$Posterior/r$	0.5347	0.2458	0.5145	0.5691	0.1258	0.5282
$1/r$	0.5142	0.2302	0.4939	0.5640	0.1338	0.5265

Table 7. MTWV comparison for various scoring strategies for Tagalog FP.

Recently, it has been shown that the best STD performance is obtained by combining systems using diverse ASR models [17]. The proposed architecture is especially beneficial for such approaches. For a given set of queries, after the one-time conversion of the OOV queries to IV sequences, the only remaining step is a composition of this FST with the small word CN index for each ASR system. In comparison, each phone-level OOV FST is composed with the large phone-level index corresponding to each ASR component for the baseline lattice approach, which is a time-consuming process. For a 5-system combination for the Tagalog FP system, we reduced the total indexing time from 43 hours to 30 minutes, and the search time from 90 hours to 3 hours. By eliminating all the phonetic indexes for the 5 ASR systems and replacing the word lattice indexes with the much smaller CN indexes, we obtain a index which is 25 times smaller than the original.

6. CONCLUSION AND FUTURE WORK

We describe a WFST STD architecture in which a word index created from confusion networks is sufficient for high-performance open vocabulary term retrieval. For each OOV term we find the sequences of IV words which could substitute it in the search process. In this paper we used phone confusability transducer as the vehicle for query expansion, although this could be replaced with any other sub-word confusability transducer. The resulting index is very small while improving performance on a variety of languages and conditions. For languages with unreliable word segmentation (Cantonese, Vietnamese, etc), there is a simple extension to the current indexing technique to be able to retrieve any subpart of a hypothesized word in a CN. We will address this category of languages in future work.

Acknowledgement

We are grateful to Jia Cui, Bhuvana Ramabhadran and Xiaodong Cui of IBM Research for building the ASR system and phone confusability model used in these experiments. This effort uses the IARPA Babel Program base period language collection releases babel104b-v0.4bY, babel105b-v0.4 and babel106b-v0.2g. 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.

7. REFERENCES

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

[2] C. Chelba, T.J. Hazen, and M. Saraçlar, "Retrieval and browsing of spoken content," *IEEE Signal Processing Magazine*, vol. 25, no. 3, pp. 39–49, 2008.

[3] D. Can and M. Saraclar, "Lattice indexing for spoken term detection," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2338–2347, 2011.

[4] M. Saraclar and R. Sproat, "Lattice-based search for spoken utterance retrieval," in *Proc. HLT-NAACL*, 2004, pp. 129–136.

[5] C. Parada, A. Sethy, and B. Ramabhadran, "Query-by-example spoken term detection for OOV terms," in *Proc. ASRU*, 2009, pp. 404–409.

[6] P. Yu and F. Seide, "A hybrid word/phoneme-based approach for improved vocabulary-independent search in spontaneous speech," in *Proc. Interspeech*, 2004.

[7] T.J. Hazen T. Hori, I.L. Hetherington and J.R. Glass, "Open-vocabulary spoken utterance retrieval using confusion networks," in *Proc. ICASSP*, 2007.

[8] V. Turunen and M. Kurimo, "Indexing confusion networks for morph based spoken document retrieval," in *Proc. SIGIR*, 2007.

[9] K. Vertanen, "Combining open vocabulary recognition and word confusion networks," in *Proc. ICASSP*, 2008.

[10] S. Nakagawa, K. Iwami, Y. Fujii, and K. Yamamoto, "A robust/fast spoken term detection method based on a syllable n-gram index with a distance metric," 2013, vol. 55, pp. 470–485.

[11] M. Akbacak, D. Vergyri, and A. Stolcke, "Open-vocabulary spoken term detection," in *Proc. Interspeech*, 2007, pp. 2393–2396.

[12] I. Szoke, L. Burget, J. Cernock, and M. Fapo, "Sub-word modeling of out of vocabulary words in spoken term detection," in *Proc. IEEE Workshop on Spoken Language Technology*, 2008.

[13] D. Can, E. Cooper, A. Sethy, C. White, B. Ramabhadran, and M. Saraclar, "Effect of pronunciation on OOV queries for spoken term detection," in *Proc. ICASSP*, 2009, pp. 3957–3960.

[14] U. Chaudhari and M. Picheny, "Matching criteria for vocabulary-independent search," in *IEEE Transactions on Audio Speech and Language Processing*, 2012, vol. 20, pp. 1633–1642.

[15] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proc. SIGIR*, 2007, pp. 615–622.

[16] Z. Xhou, P. Yu, C. Chelba, and F. Seide, "Towards spoken-document retrieval for the internet: Lattice indexing for large-scale web-search architectures," in *Proc. HLT*, 2006.

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

[18] L.Mangu and M. Padmanabhan, "Error corrective mechanisms for speech recognition," in *Proc. ICASSP*, 2001.

[19] "IARPA broad agency announcement IARPA-BAA-11-02," 2011.

[20] NIST, "The spoken term detection (STD) 2006 evaluation plan.," in *http://www.nist.gov/speech/tests/std/*, 2006.

[21] H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila speech recognition toolkit," in *Proc. SLT*, 2010, pp. 97–102.

[22] B. Kingsbury, T.N. Sainath, and H. Soltau, "Scalable minimum Bayes risk training of deep neural network acoustic models using distributed Hessian-free optimization," in *Proc. Interspeech*, 2012.

[23] H. Soltau and G. Saon, "Dynamic network decoding revisited," in *Proc. ASRU*, 2009.

[24] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "OpenFST: A general and efficient weighted finite-state transducer library," in *Proc. CIAA*, 2007, pp. 11–23.

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

[26] L. Mangu, H. Soltau, H.-K. Kuo, and G. Saon, "The IBM keyword search system for the DARPA RATS program," in *Proc. ASRU*, 2013.