INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Team ELISA System for DARPA LORELEI Speech Evaluation 2016

Pavlos Papadopoulos[1], *Ruchir Travadi*[1], *Colin Vaz*[1], *Nikolaos Malandrakis*[1], *Ulf Hermjakob*[2], *Nima Pourdamghani*[2], *Michael Pust*[2], *Boliang Zhang*[3], *Xiaoman Pan*[3], *Di Lu*[3], *Ying Lin*[3], *Ondřej Glembek*[4], *Murali Karthick Baskar*[4], *Martin Karafiát*[4], *Lukáš Burget*[4], *Mark Hasegawa-Johnson*[5], *Heng Ji*[3], *Jonathan May*[2], *Kevin Knight*[2], *Shrikanth Narayanan*[1]

[1]Signal Analysis and Interpretation Lab, University of Southern California, U.S.A.
[2]Information Sciences Institute, University of Southern California, U.S.A.
[3]Computer Science Department, Rensselaer Polytechnic Institute, U.S.A
[4]Speech@FIT, Brno University of Technology, Czech Republic
[5]Statistical Speech Technology Group, University of Illinois Urbana-Champaign, U.S.A.

{ppapadop, travadi, cvaz, malandra}@usc.edu, {ulf, damghani, pust}@isi.edu, {zhangb8, panx2, lud2, liny9}@rpi.edu, {glembek, baskar, karafiat, burget}@fit.vutbr.cz, jhasegaw@illinois.edu, jih@rpi.edu, {jonmay, knight}@isi.edu, shri@sipi.usc.edu

Abstract

In this paper, we describe the system designed and developed by team ELISA for DARPA's LORELEI (Low Resource Languages for Emergent Incidents) pilot speech evaluation. The goal of the LORELEI program is to guide rapid resource deployment for humanitarian relief (e.g. for natural disasters), with a focus on "low-resource" language locations, where the cost of developing technologies for automated human language tools can be prohibitive both in monetary terms and timewise. In this phase of the program, the speech evaluation consisted of three separate tasks: detecting presence of an incident, classifying incident type, and classifying incident type along with identifying the location where it occurs. The performance metric was area under curve of precision-recall curves. Team ELISA competed against five other teams and won all the subtasks.

1. Introduction

Efficient and timely resolution of emergency incidents is of critical importance for those affected. Collecting and analyzing information regarding those incidents is essential for providing the appropriate response. This task, though challenging in itself, becomes more complicated when emergencies occur in locations where language resources and tools are scarce. Hence, rapid creation or adaptation of technologies for information extraction on low-resource languages is crucial to guide relief efforts. DARPA's LORELEI program [1] aims to facilitate development of human language technologies for low-resource languages, with a focus on emerging volatile situations like natural disasters, food shortage, etc, and assist in the deployment of humanitarian relief teams or resources by providing situational awareness.

In the LORELEI program, situational awareness is represented through Situation Frames (SF) [2]. In the pilot speech evaluation task, a SF contains information regarding the type of incident and location. In this phase of the program, there are 11 different **types** of interest: *Evacuation*, *Food Supply*, *Urgent Rescue*, *Utilities-Energy-Sanitation*, *Infrastructure*, *Medical Assistance*, *Shelter*, *Water Supply*, *Civil Unrest-Widespread Crime*, *Elections-Politics*, and *Terrorism-Extreme Violence*. Information about location might not be available for the incident of the frame. A "document" may contain zero or multiple SFs. For the purposes of the pilot speech evaluation task, a document is an audio clip (no more than than 2 minutes) with speech in some low-resource language of interest.

APPEN [3] collected and annotated data from multiple languages. The process resulted in data packs for 7 languages, 5 for training purposes and 2 for evaluation, each corresponding to roughly fourteen hours of audio. The training languages were Amharic, Hausa, Russian, Turkish, and Uzbek containing SF annotations and no transcriptions, subsets of which were used to train or adapt the components of our system. The evaluation languages were Mandarin Chinese, and Uyghur. Mandarin was considered an unconstrained scenario, with teams being allowed to use any resource available to them. On the other hand, Uyghur was a constrained scenario, in which teams could only use resources that were collected before the announcement of the evaluation languages, along with a parallel text corpus created by LDC [2], and resources from other languages. Moreover, in the constrained scenario, each team had access to a native informant (NI), a native Uyghur speaker, for a total of 2 hours. Although, the NI was not allowed access to the evaluation data, teams could use the NI in any other way to improve their systems. Finally, development data sets were released for both the evaluation languages during the evaluation period. These datasets included only audio, without any annotations (transcripts, SF annotations, etc).

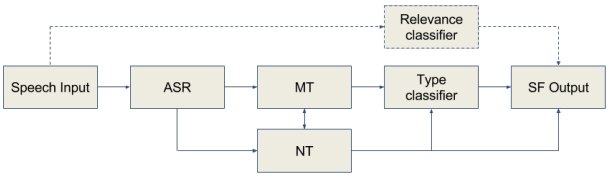

Figure 1: *General system pipeline*

Teams were required to submit results within 10 days of data release. The submitted SFs contained 4 fields, *DocumentID*, *Type*, *PlaceMention*, and *TypeConfidence*, where TypeConfidence is a confidence score [0-1] referring to the SF itself. Evaluation consisted of 3 separate tasks. The first, Relevance Classification, was a binary classification task. Given a document the system should decide if it contained at least 1 SF or not. Type Detection was the second task, in which we were asked to produce SFs whose *Type* field matched the ground

http://dx.doi.org/10.21437/Interspeech.2017-180

truth. Last, the Type+Place detection task required to produce SFs whose *Type* and *PlaceMention* fields matched the ground truth.

For every document in the evaluations sets, we had to produce SFs containing information about the type of the incident, the location, along with a confidence score. Extracting the necessary information from low-level acoustic features can be challenging given the difference in languages and the complexity of the problem. Thus, the strategy we followed was to go from audio to some level of semantic representation. To achieve this task, team ELISA built a sophisticated system combining expertise from different domains.

In Figure 1, we present the general architecture of our system. The document, containing speech in some language, goes through an Automatic Speech Recognition (ASR) component producing a transcript in that language. The transcript is processed by a Name Tagger (NT) and a Machine Translation (MT) component translating the ASR output into English. Following this, the outputs of the NT and MT components pass through a Type Classifier to produce SFs. A Type Classifier produces frames for each of the 11 types with a corresponding confidence score as well as location information, if available. Our Type Classifier accepts input in English. The reason we made this decision is the public availability of large amounts of data and corpora in English which enables us to build robust systems and check their validity instead of directly relying on information of low-resource languages. Finally, a Relevance Classifier (RC) can optionally be applied to determine if the document is in-domain or out-of-domain, i.e., if it contains information regarding any of the 11 situation types of interest or not. In the 2 evaluation languages, we made slight modifications to accommodate specific conditions. Team ELISA competed against 5 other teams and won all the subtasks for both of the evaluation languages.

The rest of the paper is organized as follows. In section 2 we describe the individual components of our system. In section 3 we present the systems for the 2 languages, and how we engineered the components for each case. In section 4 we show the results for all the tasks and finally in section 5 we draw our conclusions.

2. System Components

In this section we give an overview of the individual components presented in Figure 2.

2.1. Relevance Classification (RC)

For the RC task, we extract low-level audio features in order to generalize across different languages. We first extract OpenSmile audio features [4], which include various statistical functionals of speech properties (such as pitch, energy, and jitter) and then appended ivectors [5] to the OpenSmile features. We perform feature selection using recursive feature elimination and leave-one-language out cross-validation to discard low-value features, reducing feature dimensionality from $6,773$ to 71. Finally, we train an SVM classifier on the reduced feature set with LibSVM [6] using a 2^{nd} degree polynomial kernel.

2.2. Automatic Speech Recognition (ASR)

We use the Kaldi toolkit [7] to build the ASR systems used in our work. We built ASR systems for different languages. During the training phase the ASR outputs are used to train or adapt the rest of the components, while in the evaluation phase, ASR output goes through the rest of the pipeline to produce Situation Frames. Depending on data availability and quality of language packs the systems were built using different feature sets and models.

2.3. Name Tagging (NT)

The goal of situation frame localization is to identify a geopolitical entity (GPE) or a natural location (LOC) where a situation occurs. To achieve this goal we use bi-directional Long Short Term Memory networks (LSTMs), which can leverage long distance features with a Conditional Random Fields (CRFs) layer to capture classification dependencies [8]. NT and MT performance were used to guide the design of ASR, by enriching vocabulary and expanding the language models (LM), on both the evaluation languages.

2.4. Machine Translation (MT)

Our machine translation engine is a syntax-based statistical system that applies weighted foreign string-to-English subtree rules in order to form a fluent and adequate English translation of the input [9, 10]. The training procedure consists of extracting rules from example foreign-to-English sentence translations, upon which syntactic analysis trees are automatically induced [11], and then collecting statistics over those rules to obtain thousands of per-rule feature functions. We also use a flexible rule-based system for translating numbers, dates, and quantities, using manually constructed rules. The integration of both automatically and manually constructed rules is governed by learning the relative weights of each rule's feature functions with custom machine learning methods that optimize the non-convex BLEU (bilingual evaluation understudy) evaluation metric [12].

2.5. Type Classification (TF) and Situation Frame Production

The final component in our system pipeline is the type classifiers. We used 2 neural networks designed to be applied to English text.

The first model is a compositional topic model that accepts documents as input, uses a convolutional layer (CNN) to compose word embeddings into sentences and a forward unidirectional recurrent layer with gated units (GRU) to compose sentences into documents. This architecture is called CNN-GRU and has been used in literature for text classification [13]. GloVe word embeddings [14] were used to initialize neural network embeddings. The model was trained using about 250,000 disaster-related documents retrieved from ReliefWeb [15] and the final layer acts as 40 independent binary classifiers, each corresponding to a topic or disaster type in the ReliefWeb inventory. Application of this model to the SF task was facilitated by creating a deterministic mapping from ReliefWeb to SF categories, e.g., "Food and Nutrition" to "Food Supply".

The second model (MLP-LSA) is a bag-of-words feed-forward network with Latent Semantic Analysis (LSA) vectors as inputs. LSA vectors are produced by creating term frequency (TF) vectors, transforming to term frequency - inverse document frequency (TF-IDF) vectors and then to LSA vectors [16]. The transformation matrices for TF-IDF and LSA were learned on the ReliefWeb corpus. The actual network was initially trained on the ReliefWeb corpus, then the final layer was replaced and the entire network was re-trained with SF annotated data, hence the final layer acted as 11 independent binary classifiers.

In order to produce localized frames (frames whose PlaceMention field is not empty) we follow a simple approach: Given a detected location mention (done by the NT component), we collect all sentences containing that mention and form a "dummy" document out of them. This dummy document is

passed through the same model and generates types which are filtered by the complete document labels (i.e., a dummy document is not allowed to contain a type that was not contained in the complete document). If no entity mention is connected to a type that was detected at the document level, then a non-localized frame is created for the specific type.

Models were trained using different data based on the evaluation language. However, the following sets were common across both evaluation languages: The ReliefWeb corpus of disaster-related documents and an internal dataset of about 4000 annotated English tweets was used to train models.

3. Evaluation Systems

In the following, we present the specific system configurations which address task-specific challenges.

3.1. Mandarin Chinese System

For the Chinese Mandarin evaluation, we were allowed to use any resource available at our disposal. Hence, we were able to build two different ASR systems. The ASR outputs are processed by the rest of the pipeline producing intermediate SFs which are finally combined to provide document level SFs. This architecture is presented in Figure 2.

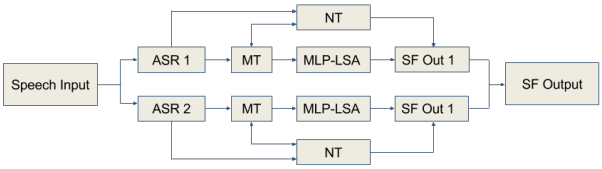

Figure 2: *Chinese Mandarin system architecture*

For this system, we use the MLP-LSA type classifier since the performance of the CNN-GRU network or their combination was found to be worse than MLP-LSA. We use the corpora described in Section 2.5 as well as the Turkish, Uzbek, and Amharic languages from the LORELEI training languages pack to train the MLP-LSA network. Since the network accepts English input, we create ASRs for those languages to decode the utterances and use MT to get English translations. However, the performance of the Amharic ASR was not satisfactory and reduced type classification performance, hence it was dropped from our training corpus. This observation indicates that a robust ASR can enhance the overall pipeline.

The MT engine is trained on a Mandarin-English corpus of 200 million words (per language), according to the procedure described in Section 2.4. The training data consisted of news text, discussion forum data, and SMS informal chat. The NT component is a Chinese name tagger [8] applied on Mandarin ASR output. To address out of vocabulary (OOV) ASR issues, we expand the vocabulary to include name gazetteers adding 17,491 entries, and a list of 112 incident related keywords derived from the Leidos corpus released by the DARPA LORELEI program and translated by bi-lingual dictionaries.

The first ASR is built using the GALE Mandarin corpus[17]. We augment this dataset through speed and volume perturbations. A Time-Delay Neural Network (TDNN) [18] with 6 hidden layers (with context windows of 1 for the first three layers, 3 for layers four and five, and 6 for the final one) handled acoustic modeling. The network is trained using the log-probability of the correct phone sequence as the objective function, also known as "chain models" [19]. The training set consists of high resolution mel-frequency cepstral coefficients (MFCCs), with labels produced from alignments generated from an HMM-GMM model trained on MFCCs and Pitch. The second ASR is trained on the HKUST corpus [20]. For acoustic modeling we use an HMM-GMM model optimizing the minimum phone error (MPE) [21] criterion on two feature streams. The first stream is represented by 52-dimensional perceptual linear predictor (PLP) features (13 PLP and up to 3rd order derivatives), which is reduced to 39 dimensions using Heteroscedastic Linear Discriminant Analysis (HLDA) [22]. The second stream was represented by 30-dimensional features produced by a Stacked Bottleneck Network (SBN) [23]. In both ASRs the lexicon is built by converting the LM character vocabulary first to Pinyin and then to SAMPA. The LMs of the two ASRs are created based on their respective corpora. We refine ASR LMs by computing the frequency of each name token in a large monolingual corpus consisted of Chinese Gigaword and TAC-KBP source collection, and then merge this name unigram LM with the original LMs by linear interpolation.

3.2. Uyghur System

For the Uyghur evaluation, our resources were highly constrained. Hence we had to change the configuration of our system, both in terms of its architecture and its design of its components. The system pipeline used for obtaining SFs on Uyghur data is presented in Figure 3. Since WER for the Uyghur ASR system was expected to be high, we decided to use the RC in order to reduce the dependence on ASR output.

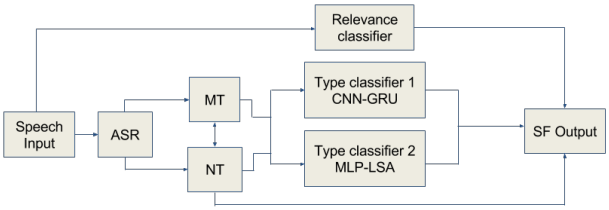

Figure 3: *Uyghur system architecture*

The RC estimates the probability $P(r = 1)$ that a document is relevant (contains at least 1 incident). The type classifiers estimate the probabilities $P(t_i = 1|r = 1)$ that a type t_i is present in the utterance, given that it is relevant. Ultimately, the probability $P(t_i = 1)$ that a type is present in an utterance is obtained as :

$$\begin{aligned} P(t_i = 1) &= P(r = 1)P(t_i = 1|r = 1) + \\ &\quad P(r = 0)P(t_i = 1|r = 0) \\ &= P(r = 1)P(t_i = 1|r = 1) \end{aligned} \quad (1)$$

where the last equality follows from the fact that a type can be present only if the segment is relevant. Therefore, the probability of occurrence of a type is just a product of outputs from the relevance and type classifiers. Since, the type classifiers used in this system estimate $P(t_i = 1|r = 1)$, they are trained only on utterances that are relevant. Moreover, we combine CNN-GRU and MLP-LSA type classifier outputs using the maximum posterior probability per output class, meaning that the final output is the union of the individual model outputs. In addition to the corpora described in Section 2.5, we used Turkish, Uzbek, and Mandarin to train our models. Turkish and Uzbek were part of the LORELEI training languages pack, and DARPA allowed us to use Mandarin for the Uyghur task. In each case, the outputs of language-specific ASRs are translated to English.

Insufficient data prevented us from developing a robust Uyghur name tagger (Uyghur speech localization F-score is 6.9%). Since Uzbek and Uyghur have phonetic and word similarities, we run an Uzbek ASR (trained on 10 hours of speech

transcribed by crowdsourcing) directly on Uyghur speech. Although, our Uzbek name tagger achieves a 84% F-score on clean Uzbek texts, only 194 name mentions were identified. To overcome this issue, we translate the ASR output into English and use an English name tagger. Through various similarity measures including string match, edit distance, soundex, gazetteers, and word alignment, we project names from English to Uzbek, increasing the number of name mentions to 1,014. Finally, we convert names from Uzbek to Uyghur according to the following procedure: First, we check if an identified Uzbek name exists in our Uzbek-English name gazetteer and its English translation exists in our Uyghur-English name gazetteer. Second, we calculate the edit distance between an Uzbek name and a romanized Uyghur name in our gazetteer and if it is less than 2 we adopt the Uyghur form. Finally, for the remaining cases we rely on a joint source-channel converter [24] trained on Uzbek-Uyghur name pairs mined from Wikipedia.

In order to train the MT engine we use a Uyghur-English corpus of 2.3 million words (per language) provided by DARPA. We also acquired $239,000$ entries by cleaning human lexicons. These entries were expanded, providing $578,000$ bilingual entries, and added in our training set. Moreover, we employ an Uzbek-to-Uyghur transliteration scheme to further increase our training corpus. This scheme was realized by exploiting vocabulary and grammatical similarities between Uzbek and Uyghur. We use Uzbek-English and Uyghur-English dictionaries and, by pivoting on English, we create an Uzbek-Uyghur dictionary. This was used to train a transliteration system [25] and apply it to the 1.8 million words of available Uzbek-English training data. Parameter optimization [12] was performed to maximize BLEU on an in-domain parallel corpus constructed by non-Uyghur speakers and validated by a native Uyghur speaker (not the NI). Once ASR outputs and name entities are acquired, we re-train our MT and re-decode the transcriptions, which are then forwarded to the type classifier.

Finally, to train the ASR we used the NI. We held 2 sessions, which were split into 2 parts: i) reading, and ii) transcribing, each lasting 30 minutes (referred to as R1, T1, R2, and T2, respectively), with T1 and T2 used for ASR internal evaluations. For the reading parts we used sentences from the LORELEI 2016 text evaluation (a previous phase of the program) and from parallel text, while for the transcribing parts we used the provided Uyghur development dataset. We combined R1, R2, and 4 hours of data obtained by transliterating Uzbek to Uyghur (U2U) to create our training set. We used data augmentation on these 3 sets by introducing speed and vocal perturbations. The ASR acoustic models follow a DNN-HMM architecture [26], specifically a feed-forward neural network with 2 hidden layers of 256 neurons. The network was trained on features obtained by splicing 7 frames of 13-dimensional PLPs appended by 69-dimensional Multilingual regional dependent transforms (RDTs) [27]. Additional feature processing was performed following the steps described in [28] to provide a label set for DNN training. We used a trigram LM prepared from Gazetteer words, an Uyghur text corpus of roughly 1.5 million utterances, and a parallel text dataset. The lexicon consisted of 58,917 unique words, whose contents were influenced by NT and MT design and was based on a direct grapheme-to-phoneme mapping.

4. Results

Since the desired operating point of a SF system is subject to change, systems were evaluated at various operating points. Area under the curve (AUC) of precision-recall (PR) curves was used as a summary statistic to rank system performance.

Teams were allowed to submit one primary (P) and two contrastive systems (C1 and C2) in both evaluation scenarios. The primary systems were used for performance ranking, and the contrastives for internal use. In the Mandarin evaluation, each of the contrastive systems produced SFs using only 1 ASR instead of their combination (Section 3.1). For Uyghur, the first contrastive system used Uzbek ASR, and the second used the same ASR as the primary (Section 3.2) but without NI input. The results for all systems are presented in Table 1.

Table 1: *System Results of team ELISA. P stands for primary, C1 and C2 for contrastive 1 and 2 respectively.*

	Mandarin			**Uyghur**		
	P	C1	C2	P	C1	C2
Relevance	0.673	0.677	0.622	0.701	0.654	0.699
Type	0.291	0.260	0.237	0.254	0.125	0.214
Type+Place	0.021	0.029	0.017	0.013	0.002	0.010

In the relevance task (which can be considered the simplest of the three), all of our systems have similar performance, per evaluation language. In Mandarin, we did not use the RC and relied on SF confidence scores to make this decision, since we had confidence on the robustness of our ASRs. In Uyghur, use of RC boosted the performance of our system.

All of our primary systems outperformed the contrastive ones for type classification. In Mandarin, C1 was using a more advanced ASR than C2 and this is reflected on SF type performance, with combination of the two ASRs providing a boost over their individual performances. The P system in Uyghur outperforms C1 and C2 for similar reasons, the ASR used in P outperformed those of C1 and C2.

Finally, the Type+Place task was the most challenging of all. Although, the underlying NT component was performing well (e.g. 84% F-score on clean Uzbek), the errors of the other components compounded heavily with NT errors. Further investigation is needed to improve our performance on this task.

Due to space limitations we are not able to provide performance information of the individual system components. After this evaluation was conducted, it also came to light that the provided SF annotations are very noisy, with large disagreements amongst annotators, which adds an additional layer of difficulty; and reflects the complexity of the task. However, team ELISA outperformed 5 other competing teams on every task for both the evaluation languages.

5. Conclusion

We presented the system developed by team ELISA for the pilot speech evaluation of LORELEI. The goal of this program is to aid humanitarian assistance by guiding the deployment of relief teams and resources. Our systems ranked first for all of the 6 tasks of this evaluation. Our experiments indicate that building a robust ASR is crucial to the whole pipeline since it directly affects the performance of the other components. We continue to investigate how to enhance the performance of the individual components, and explore ways to reduce error effects of components transferring to the overall system.

6. Acknowledgements

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-15-C-0115.

7. References

[1] "DARPA LORELEI website," http://www.darpa.mil/program/low-resource-languages-for-emergent-incidents.

[2] S. Strassel and J. Tracey, "Lorelei language packs: Data, tools, and resources for technology development in low resource languages," in *Proceedings of LREC*, 2016, pp. 3273–3280.

[3] "APPEN website," http://appen.com/.

[4] F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in opensmile, the munich open-source multimedia feature extractor," in *Proceedings of the 21st ACM International Conference on Multimedia*, ser. MM '13, 2013, pp. 835–838.

[5] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 2, pp. 788–798, 2011.

[6] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 19, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[7] 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*.

[8] D. Yu, X. Pan, B. Zhang, L. Huang, D. Lu, S. Whitehead, and H. Ji, "RPI_BLENDER TAC-KBP2016 system description," in *Proceedings of the 2016 Text Analysis Conference (TAC2016)*.

[9] M. Galley, M. Hopkins, K. Knight, and D. Marcu, "What's in a translation rule?" in *HLT-NAACL 2004: Main Proceedings*, D. M. Susan Dumais and S. Roukos, Eds. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2 - May 7 2004, pp. 273–280.

[10] M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, and I. Thayer, "Scalable inference and training of context-rich syntactic translation models," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, 2006, pp. 961–968.

[11] D. M. Bikel, "Design of a multi-lingual, parallel-processing statistical parsing engine," in *HLT*, 2002.

[12] D. Chiang, K. Knight, and W. Wang, "11,001 new features for statistical machine translation," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, ser. NAACL '09, 2009, pp. 218–226.

[13] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, pp. 2267–2273.

[14] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2014, pp. 1532–1543.

[15] "Reliefweb. retrieved march 31, 2016," http://reliefweb.int/.

[16] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[17] "GALE Phase 2 Chinese Broadcast News Speech, LDC2013S08," https://catalog.ldc.upenn.edu/LDC2013S08.

[18] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *INTERSPEECH, 16th Annual Conference of the International Speech Communication Association*, 2015, pp. 3214–3218.

[19] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *INTERSPEECH, 17th Annual Conference of the International Speech Communication Association*, 2016, pp. 2751–2755.

[20] "HKUST Mandarin Telephone Speech," https://catalog.ldc.upenn.edu/LDC2005S15.

[21] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, University of Cambridge, 2003.

[22] N. Kumar and A. G. Andreou, "Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition," *Speech Communication*, vol. 26, pp. 283–297, 1998.

[23] F. Grezl, M. Karafiat, and L. Burget, "Investigation into bottle-neck features for meeting speech recognition," in *Proc.INTERSPEECH, Annual Conference of the International Speech Communication Association*, 2009, pp. 2947–2950.

[24] Y. Lin, X. Pan, A. Deri, H. Ji, and K. Knight, "Leveraging entity linking and related language projection to improve name transliteration," in *In Proceedings of ACL2016 Workshop on Named Entities*, 2016.

[25] S. Jiampojamarn, C. Cherry, and G. Kondrak, "Joint processing and discriminative training for letter-to-phoneme conversion," in *In Proceedings of ACL*, 2008, pp. 905–913.

[26] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[27] M. Karafiát, L. Burget, F. Grézl, K. Veselý, and J. H. Černocký, "Multilingual region-dependent transforms," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 5430–5434.

[28] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative Training of Deep Neural Networks," in *Proceedings of Interspeech 2013*, 2013, pp. 2345–2349.