



## Topic Identification for Speech without ASR

Chunxi Liu<sup>1</sup>, Jan Trmal<sup>1,2</sup>, Matthew Wiesner<sup>1</sup>, Craig Harman<sup>1</sup>, Sanjeev Khudanpur<sup>1,2</sup>

<sup>1</sup>Center for Language and Speech Processing, The Johns Hopkins University, USA

<sup>2</sup>Human Language Technology Center of Excellence, The Johns Hopkins University, USA

{chunxi, yenda, wiesner, khudanpur}@jhu.edu, craig@craigharman.net

### Abstract

Modern topic identification (topic ID) systems for speech use automatic speech recognition (ASR) to produce speech transcripts, and perform supervised classification on such ASR outputs. However, under resource-limited conditions, the manually transcribed speech required to develop standard ASR systems can be severely limited or unavailable. In this paper, we investigate alternative unsupervised solutions to obtaining tokenizations of speech in terms of a vocabulary of automatically discovered word-like or phoneme-like units, without depending on the supervised training of ASR systems. Moreover, using automatic phoneme-like tokenizations, we demonstrate that a convolutional neural network based framework for learning spoken document representations provides competitive performance compared to a standard bag-of-words representation, as evidenced by comprehensive topic ID evaluations on both single-label and multi-label classification tasks.

**Index Terms:** topic identification, unsupervised term discovery, acoustic unit discovery, convolutional neural networks

### 1. Introduction

Topic identification (topic ID) on speech aims to identify the topic(s) for given speech recordings, referred to as spoken documents, where the topics are a predefined set of classes or labels. This task is typically formulated as a three-step process. First, speech is tokenized into words or phones by automatic speech recognition (ASR) systems [1], or by limited-vocabulary keyword spotting [2]. Second, standard text-based processing techniques are applied to the resulting tokenizations, and produce a vector representation for each spoken document, typically a bag-of-words multinomial representation, or a more compact vector given by probabilistic topic models [3, 4]. Finally, topic ID is performed on the spoken document representations by supervised training of classifiers, such as Bayesian classifiers and support vector machines (SVMs).

However, in the first step, training the ASR system required for tokenization itself requires transcribed speech and pronunciations. In this paper, we focus on a difficult and realistic scenario where the speech corpus of a test language is annotated only with a minimal number of topic labels, i.e., no manual transcriptions or dictionaries for building an ASR system are available. We aim to exploit approaches that enable topic ID on speech without any knowledge of that language other than the topic annotations.

In this scenario, while previous work demonstrates that the cross-lingual phoneme recognizers can produce reasonable speech tokenizations [5, 6], the performance is highly dependent on the language and environmental condition (chan-

nel, noise, etc.) mismatch between the training and test data. Therefore, we focus on unsupervised approaches that operate directly on the speech of interest. Raw acoustic feature-based unsupervised term discovery (UTD) is one such approach that aims to identify and cluster repeating word-like units across speech based around segmental dynamic time warping (DTW) [7, 8]. [9] shows that using the word-like units from UTD for spoken document classification can work well; however, the results in [9] are limited since the acoustic features on which UTD is performed are produced by acoustic models trained from the transcribed speech of its evaluation corpus. In this paper, we investigate UTD-based topic ID performance when UTD operates on language-independent speech representations extracted from multilingual bottleneck networks trained on languages other than the test language [10]. Another alternative to producing speech tokenizations without language dependency is the model-based approach, i.e., unsupervised learning of hidden Markov model (HMM) based phoneme-like units from untranscribed speech. We exploit the Variational Bayesian inference based acoustic unit discovery (AUD) framework in [11] that allows parallelized large-scale training. In topic ID tasks, such AUD-based systems have been shown to outperform other systems based on cross-lingual phoneme recognizers [6], and this paper aims to further investigate how the performance compares among UTD, AUD and ASR based systems.

Moreover, after the speech is tokenized, these works [1, 2, 5, 6, 9, 10] are limited to using bag-of-words features as spoken document representations. While UTD only identifies relatively long (0.5 – 1 sec) repeated terms, AUD/ASR enables full-coverage segmentation of continuous speech into a sequence of units/words, and such a resulting temporal sequence enables another feature learning architecture based on convolutional neural networks (CNNs) [12]; instead of treating the sequential tokens as a bag of acoustic units or words, the whole token sequence is encoded as concatenated continuous vectors, and followed by convolution and temporal pooling operations that capture the local and global dependencies. Such continuous space feature extraction frameworks have been used in various language processing tasks like spoken language understanding [13, 14] and text classification [15, 16]. However, three questions are worth investigating in our AUD-based setting: (i) if such a CNN-based framework can perform as well on noisy automatically discovered phoneme-like units as on orthographic words/characters, (ii) if pre-trained vectors of phoneme-like units from *word2vec* [17] provide superior performance to random initialization as evidenced by the word-based tasks, and (iii) if CNNs are still competitive in low-resource settings of hundreds to two-thousand training exemplars, rather than the large/medium sized datasets as in previous work [15, 16].

Finally, incorporating the different tokenization and feature representation approaches noted above, we perform com-

This work was partially supported by DARPA LORELEI Grant No HR0011-15-2-0024, NSF Grant No CRI-1513128, and IARPA Contract No 2012-12050800010.

prehensive topic ID evaluations on both single-label and multi-label spoken document classification tasks.

## 2. Unsupervised tokenizations of speech

### 2.1. Unsupervised term discovery (UTD)

UTD aims to automatically identify and cluster repeated terms (e.g. words or phrases) from speech. To circumvent the exhaustive DTW-based search limited by  $\mathcal{O}(n^2)$  time [7], we exploit the scalable UTD framework in the Zero Resource Toolkit (ZRTools) [8], which permits search in  $\mathcal{O}(n \log n)$  time. We briefly describe the UTD procedures in ZRTools by four steps below, and full details can be found in [8].

1. Construct the sparse approximate acoustic similarity matrices between pairs of speech utterances.
2. Identify word repetitions via fast diagonal line search and segmental DTW.
3. The resulting matches are used to construct an acoustic similarity graph, where nodes represent the matching acoustic segments and edges reflect DTW distances.
4. Threshold the graph edges, and each connected component of the graph is a cluster of acoustic segments, which produces a corresponding term (word/phrase) category.

Finally, the cluster of each discovered term category consists of a list of term occurrences.

Note that in the third step above, the weight on each graph edge can be exact DTW-based similarity, or other similarity based on heuristics more than DTW distance. For example, we investigate an implementation in ZRTools, where a separate logistic regression model is used to rescore the similarity between identified matches by determining how likely the matching pair is the same underlying word/phrase and is not a filled pause (e.g. “um-hum” and “yeah uh-huh” in English). Filled pauses tend to be acoustically stationary with more phone repeats and thus would match throughout the acoustic similarity matrix, whereas a contentful word (without too many phone repeats) tend to concentrate around the main diagonal; thus, the features in logistic regression contain the numbers of matrix elements in diagonal bands in progressive steps away from the main diagonal. Feature weights are learned using a portion of transcribed speech with reference transcripts, and the resulting model can be used for language-independent rescoring.

### 2.2. Acoustic unit discovery (AUD)

We exploit the nonparametric Bayesian AUD framework in [11] based on variational inference, rather than the maximum likelihood training in [5] which may oversimplify the parameter estimations, nor the Gibbs Sampling training in [19] which is not amenable to large scale applications. Specifically, a phone-loop model is formulated where each phoneme-like unit is modeled as an HMM with a Gaussian mixture model of output densities (GMM-HMM). Under the Dirichlet process framework, we consider the phone loop as an infinite mixture of GMM-HMMs, and the mixture weights are based on the stick-breaking construction of Dirichlet process. The infinite number of units in the mixture is truncated in practice, giving zero mixture weight to any unit beyond some large count. We treat such mixture of GMM-HMMs as a single unified HMM and thus the segmentation of the data is performed using standard forward-backward algorithm. Training is fully unsupervised and parallelized; after a fixed number of training iterations, we use Viterbi decoding algorithm to obtain acoustic unit tokenizations of the data.

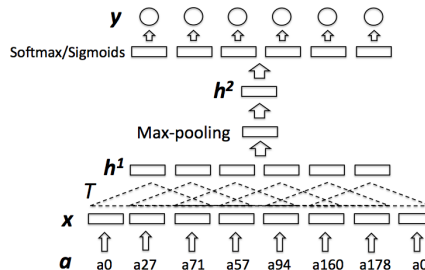


Figure 1: CNN-based framework that operates on automatically discovered acoustic units.

## 3. Learning document representations

### 3.1. Bag-of-words representation

After we obtain the tokenizations of speech by either UTD or AUD, each spoken document is represented by a vector of unigram occurrence counts over discovered terms, or a vector of  $n$ -gram counts over acoustic units, respectively. Each feature vector can be further scaled by inverse document frequency (IDF), producing a TF-IDF feature.

### 3.2. Convolutional neural network-based representation

AUD enables full-coverage tokenization of continuous speech into a sequence of acoustic units, which we can exploit in a CNN-based framework to learn a vector representation for each spoken document. As shown in Figure 1, in an acoustic unit sequence  $\mathbf{a}$  of length  $m$ , each unit  $a_i$ ,  $1 \leq i \leq m$ , is encoded as a fixed dimensional continuous vector, and the whole sequence  $\mathbf{a}$  is represented as a concatenated vector  $\mathbf{x}$ . A shared convolutional feature transform  $T$  spans a fixed-sized  $n$ -gram window,  $n \ll m$ , and slides over the whole sequence. Then the hidden feature layer  $\mathbf{h}^1$  with nonlinearities consists of each feature vector  $h_i^1$  extracted from the shared convolutional window centered at each acoustic unit position  $i$ . Max-pooling is performed on top of each  $h_i^1$ ,  $1 \leq i \leq m$ , to obtain a fixed-dimensional vector representation for the whole sequence  $\mathbf{a}$ , i.e., a vector representation of the whole spoken document, followed by another hidden layer  $\mathbf{h}^2$  and a final output layer. Note that this framework needs supervision for training; e.g., the output layer can be a softmax function for single-label classification, and the whole model is trained with categorical cross-entropy loss.

Also, the vector representation of each unique acoustic unit can be randomly initialized, or pre-trained from other tasks. Specifically, we apply the *skip-gram* model of word2vec [22] to pre-train one embedding vector for each acoustic unit, based on the hierarchical softmax with Huffman codes.

## 4. Supervised document classification

### 4.1. Single-label classification

For the bag-of-words representation, we use a stochastic gradient descent (SGD) based linear SVM [20, 21] with hinge loss and  $\mathcal{L}^1/\mathcal{L}^2$  norm regularization. For the CNN-based framework, we use a softmax function in the output layer for classification as described in Section 3.2.

### 4.2. Multi-label classification

In the setting where each spoken document can be associated with multiple topics/labels, we proceed to perform a multi-label classification task. The baseline approach is the binary rele-

vance method, which independently trains one binary classifier for each label, and the spoken document is evaluated by each classifier to determine if the respective label applies to it. Specifically, we use a set of SVMs (Section 4.1), one for each label, on the bag-of-words features.

To adapt the CNN-based framework for multi-label classification, we replace the softmax in the output layer with a set of sigmoid output nodes, one for each label, as shown in Figure 1. Since a sigmoid naturally provides output values between 0 and 1, we train the neural network (NN) to minimize the binary cross entropy loss defined as  $l(\Theta, (\mathbf{x}, \mathbf{y})) = -\sum_{k=1}^K (y_k \log o_k + (1 - y_k) \log(1 - o_k))$ , where  $\Theta$  denotes the NN parameters,  $\mathbf{x}$  is the feature vector of acoustic unit sequence,  $\mathbf{y}$  is the target vector of labels,  $o_k$  and  $y_k$  are the output and the target for label  $k$ , and the number of unique labels is  $K$ .

## 5. Experiments

### 5.1. Single-label classification

#### 5.1.1. Experimental setup

For our single-label topic classification experiments, we use the Switchboard Telephone Speech Corpus [23], a collection of two-sided telephone conversations. We use the same development (dev) and evaluation (eval) data sets as in [9, 10]. Each whole conversation has two sides and one single topic, and topic ID is performed on each individual-side speech (i.e., each side is seen as one single spoken document). In the 35.7 hour dev data, there are 360 conversation sides evenly distributed across six different topics (recycling, capital punishment, drug testing, family finance, job benefits, car buying), i.e., each topic has equal number of 60 sides. In the 61.6 hour eval data, there are another different six topics (family life, news media, public education, exercise/fitness, pets, taxes) evenly distributed across 600 conversation sides. Algorithm design choices are explored through experiments on dev data. We use manual segmentations provided by the Switchboard corpus to produce utterances with speech activity, and UTD and AUD are operating only on those utterances.

For UTD, we use the ZRTools [8] implementation with the default parameters except that, we use cosine similarity threshold  $\delta = 0.5$ , and vary the diagonal median filter duration  $\kappa$  over  $\{0.6, 0.7\}$ ; we try both the exact DTW-based similarity and the rescored similarity as described in Section 2.1, and tune the similarity threshold (used to partition the graph edges) over  $\{0.85, 0.88, 0.90, 0.92\}$ . For AUD, the unsupervised training is performed only on the dev data (10 iterations); after training, we use the learned models to decode both dev and eval data set, and obtain the acoustic unit tokenizations. We use truncation level 200, which implies maximum 200 different acoustic units can be learned from the corpus. For each acoustic unit, we use a 3-state HMM with 2 Gaussians per state. For the stick-breaking construction of Dirichlet process, we vary the concentration parameter  $\gamma$  over  $\{1.0, 10.0\}$ , and other hyperparameters are the same as [11].

The acoustic features on which UTD and AUD operate are extracted using the same multilingual bottleneck (BN) network as described in [10] with Kaldi toolkit [24]. We conduct the multilingual BN training with 10 language collections (Assamese, Bengali, Cantonese, Haitian, Lao, Pashto, Tamil, Tagalog, Vietnamese and Zulu) – 10 hours of transcribed speech per language. Complete specifications can be found in [10].

For SVM-based classification, we use the bag of discovered term unigrams, or bag of acoustic unit trigrams. On dev data, we

try using the features of raw counts or scaled by IDF, SVM regularization tuned over  $\mathcal{L}^1/\mathcal{L}^2$  norm, regularization constant tuned over  $\{0.001, 0.0001\}$ , and SGD epochs tuned over  $\{30, 50\}$ . We further normalize each feature to  $\mathcal{L}^2$  norm unit length. Each experiment is a run of 10-fold cross validation (CV) on the 360 conversation sides of dev data, or on the 600 sides of eval data, respectively. Note that our data size here is relatively small (only 360 or 600) and the SGD training may give high variance in the performance [25]. Therefore, to report classification accuracy for each configuration (when varying features or models), we repeat each CV experiment 5 times, where each experiment again is a run of 10-fold CV; then for each configuration, the mean and standard deviation of 5 experiments is reported.

For CNN-based classification, we use the same strategy to report classification accuracy, i.e., repeating experiments 5 times (where each time is a 10-fold CV) for each CNN configuration. Note that the respective 10 folds of both dev and eval data sets are fixed the same for all the SVM and CNN experiments. Additionally, for each 10-fold CV experiment, instead of training on 9 folds and testing on the remaining 1 fold as in SVM, we use 8 folds for CNN training, leave another 1 fold out as validation data; after training each CNN model for up to 100 epochs, the model with the best accuracy on the validation data is used for evaluation on the test set. The acoustic unit sequence (as CNN inputs) are zero-padded to the longest length in each dataset. We implemented the CNNs in Keras [26] with Theano [27] backend. CNN architectures are determined through experiments on dev data. For SGD training we use the Adadelta optimizer [28] and mini-batch size 18. The  $n$ -gram window size of each convolutional feature transform  $T$  is 7. The size of each hidden feature vector  $h_i^1$  (extracted from the transform  $T$ ) is 1024, with rectified linear unit (ReLU) nonlinearities. Thus, after max-pooling over time, we have a 1024-dimensional vector again, which then goes through another hidden layer  $h^2$  (also set as 1024-dimensional with ReLU) and finally into a softmax. Dropout [29] rate 0.2 is used at each layer.

When we initialize the vector representation of each acoustic unit with a set of pre-trained vectors (instead of random initializations), we apply the skip-gram model of word2vec [22] to the acoustic unit tokenizations of each data set. We use the *gensim* implementation [30], which includes a vector space of embedding dimension 50 (tuned over  $\{50, 80\}$ ), a skip-gram window of size 5, and SGD over 20 epochs.

#### 5.1.2. Results on Switchboard

Table 1 shows the topic ID results on Switchboard. For UTD-based classifications, we find that the default rescoring in ZRTools [8] which is designed to filter out the filled pauses produces comparable performance to the raw DTW similarity scores, but the rescoring can result in much faster connected-component clustering (Section 2.1). Note that this rescoring model is estimated using a portion of transcribed Switchboard, but it is still a legitimate language-independent UTD approach while operating on languages other than English. While a diagonal median filter duration  $\kappa$  of 0.6 or 0.7 gives similar results,  $\kappa = 0.7$  produces longer but fewer terms, giving more sparse feature representations. Therefore, we proceed with rescoring and  $\kappa = 0.7$  in the following UTD experiments (Section 5.2).

For AUD-based classifications, CNN without word2vec pre-training usually gives comparable results with SVM; however, using word2vec pre-training, CNN substantially outperforms the competing SVM in all cases. Also as the concentration parameter  $\gamma$  in AUD increases from 1.0 to 10.0 (yielding

Table 1: *Single-label topic ID accuracies on Switchboard.*

Dataset	Feature	Model	Accuracy
Dev	UTD	SVM	0.863 ± 0.010
	UTD w/ rescoring	SVM	0.876 ± 0.008
	AUD, # units 184	SVM	0.682 ± 0.007
		CNN	0.657 ± 0.017
	AUD, # units 199	CNN w/ word2vec	0.728 ± 0.011
		SVM	0.686 ± 0.005
Eval	UTD	SVM	0.851 ± 0.003
	UTD w/ rescoring	SVM	0.875 ± 0.003
	AUD, # units = 184	SVM	0.710 ± 0.005
		CNN	0.708 ± 0.013
	AUD, # units = 199	CNN w/ word2vec	0.762 ± 0.007
		SVM	0.700 ± 0.005
AUD, # units = 199	CNN	0.690 ± 0.015	
	CNN w/ word2vec	<b>0.767 ± 0.013</b>	

less concentrated distributions), we have more unique acoustic units in the tokenizations of both data sets, from 184 to 199, and  $\gamma = 10.0$  usually produces better results than  $\gamma = 1.0$ .

## 5.2. Multi-label classification

### 5.2.1. Experimental setup

We further evaluate our topic ID performance on the speech corpora of three languages released by the DARPA LORELEI (Low Resource Languages for Emergent Incidents) Program. For each language there are a number of audio speech files, and each speech file is cut into segments of various lengths (up to 120 seconds). Each speech segment is seen as either in-domain or out-of-domain. In-domain data is defined as any speech segment relating to an incident or incidents, and in-domain data will fall into a set of domain-specific categories; these categories are known as situation types, or in-domain topics. There are 11 situation types: “Civil Unrest or Wide-spread Crime”, “Elections and Politics”, “Evacuation”, “Food Supply”, “Urgent Rescue”, “Utilities, Energy, or Sanitation”, “Infrastructure”, “Medical Assistance”, “Shelter”, “Terrorism or other Extreme Violence”, and “Water Supply”. We consider “Out-of-domain” as the 12th topic label, so each speech segment either corresponds to one or multiple in-domain topics, or is “Out-of-domain”. We use the average precision (AP, equal to the area under the precision-recall curve) as the evaluation metric, and report both the AP across the overall 12 labels, and the AP across 11 situation types, as shown in Table 2. For each configuration, only a single 10-fold CV result is reported, since we observe less variance in results here than in Switchboard. We have 16.5 hours in-domain data and 8.5 hours out-of-domain data for Turkish, 2.9 and 13.2 hours for Uzbek, and 7.7 and 7.2 hours for Mandarin. We use the same CNN architecture as on Switchboard but make the changes as described in Section 4.2. Also we use mini-batch size 30 and fix the training epochs as 100. All CNNs use word2vec pre-training. Additionally, we also implement another two separate topic ID baselines using the decoded word outputs from two supervised ASR systems, trained from 80 hours transcribed Babel Turkish speech [31] and about 170 hours transcribed HKUST Mandarin telephone speech (LDC2005T32 and LDC2005S15), respectively.

Table 2: *Multi-label topic ID average precision on LORELEI languages, with the number of speech segments in parentheses.*

Dataset	Feature	Model	Overall	In-domain topics
Turkish (2095)	UTD	SVM	0.583	0.531
	AUD	SVM	0.627	0.556
	AUD	CNN	0.641	0.564
	ASR	SVM	0.625	0.580
Uzbek (1416)	UTD	SVM	0.803	0.254
	AUD	SVM	0.791	0.203
	AUD	CNN	0.807	0.207
Mandarin (724)	UTD	SVM	0.444	0.234
	AUD	SVM	0.436	0.220
	AUD	CNN	0.420	0.183
	ASR	SVM	0.461	0.261

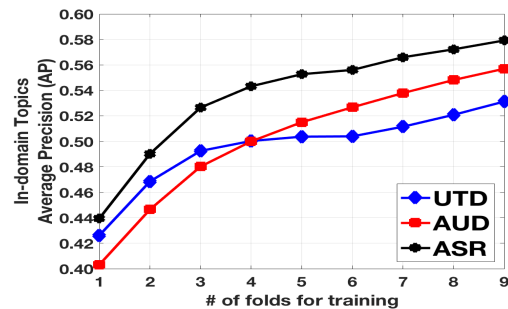


Figure 2: *Average precision of in-domain situation types on Turkish when varying the number of folds used for training.*

### 5.2.2. Results on LORELEI Datasets

As shown in Table 2, UTD-based SVMs are more competitive than AUD-based SVMs on the smaller corpora, i.e., Uzbek and Mandarin, while being less competitive on the larger corpus, Turkish. We further investigate this behavior on each individual language by varying the amount of training data; we split the data into 10 folds, and perform 10-fold CV 9 times, varying the number of folds for training from 1 to 9. As illustrated in Figure 2 for Turkish, as we use more folds for training, AUD-based system starts to be more competitive than UTD. Supervised ASR-based systems still give the best results in various cases, while UTD and AUD based systems give comparable performance.

Note that CNN-based systems outperform SVMs on Turkish and Uzbek while losing on the smaller sized Mandarin, indicating more topic-labeled data is needed to enable competitive CNNs. This also indicates why CNNs on LORELEI corpora do not produce as large a gain over SVMs as on the larger sized Switchboard, since each 15-25 hour LORELEI corpus with 12 topic labels is a relatively small amount of data compared to the 35.7/61.6 hour Switchboard corpus with 6 labels.

## 6. Concluding Remarks

We have demonstrated that both UTD and AUD are viable technologies for producing effective tokenizations of speech that enable topic ID performance comparable to using standard ASR systems, while effectively removing the dependency on transcribed speech required by the ASR alternative. We find that when training data is severely limited the UTD-based classification is superior to AUD-based classification. As the amount of training data increases, performance improves across the board. Finally, with sufficient training data AUD-based CNNs with word2vec pre-training outperform AUD-based SVMs.

## 7. References

- [1] T. J. Hazen, “MCE Training Techniques for Topic Identification of Spoken Audio Documents,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2451–2460, Nov 2011.
- [2] J. Wintrode and S. Khudanpur, “Limited resource term detection for effective topic identification of speech,” in *Proc. ICASSP*, 2014.
- [3] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.
- [4] C. May, F. Ferraro, A. McCree, J. Wintrode, D. Garcia-Romero, and B. Van Durme, “Topic identification and discovery on text and speech,” in *Proc. EMNLP*, 2015.
- [5] M.-h. Siu, H. Gish, A. Chan, W. Belfield, and S. Lowe, “Unsupervised training of an HMM-based self-organizing unit recognizer with applications to topic classification and keyword discovery,” *Computer Speech & Language*, vol. 28, no. 1, pp. 210–223, 2014.
- [6] S. Kesiraju, R. Pappagari, L. Ondel, L. Burget, N. Dehak, S. Khudanpur, J. Černocký, and S. Gangashetty, “Topic identification of spoken documents using unsupervised acoustic unit discovery,” in *Proc. ICASSP*, 2017.
- [7] A. S. Park and J. R. Glass, “Unsupervised pattern discovery in speech,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 186–197, 2008.
- [8] A. Jansen and B. Van Durme, “Efficient spoken term discovery using randomized algorithms,” in *Proc. ASRU*, 2011, <https://github.com/arenjansen/ZRTools>.
- [9] M. Dredze, A. Jansen, G. Coppersmith, and K. Church, “NLP on spoken documents without ASR,” in *Proc. EMNLP*, 2010.
- [10] C. Liu, J. Yang, M. Sun, S. Kesiraju, A. Rott, L. Ondel, P. Ghahremani, N. Dehak, L. Burget, and S. Khudanpur, “An empirical evaluation of zero resource acoustic unit discovery,” in *Proc. ICASSP*, 2017.
- [11] L. Ondel, L. Burget, and J. Černocký, “Variational inference for acoustic unit discovery,” in *Proc. SLTU*, 2016.
- [12] R. Collobert and J. Weston, “A unified architecture for natural language processing: Deep neural networks with multitask learning,” in *Proc. ICML*, 2008.
- [13] P. Xu and R. Sarikaya, “Convolutional neural network based triangular CRF for joint intent detection and slot filling,” in *Proc. ASRU*, 2013.
- [14] C. Liu, P. Xu, and R. Sarikaya, “Deep contextual language understanding in spoken dialogue systems,” in *Proc. INTERSPEECH*, 2015.
- [15] Y. Kim, “Convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1408.5882*, 2014.
- [16] X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” in *Advances in neural information processing systems*, 2015, pp. 649–657.
- [17] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.
- [18] J. Nam, J. Kim, E. L. Mencía, I. Gurevych, and J. Fürnkranz, “Large-scale multi-label text classification - revisiting neural networks,” in *Proc. ECML-PKDD*, 2014, pp. 437–452.
- [19] C.-y. Lee and J. Glass, “A nonparametric bayesian approach to acoustic model discovery,” in *Proc. ACL*, 2012.
- [20] S. Shalev-Shwartz, Y. Singer, and N. Srebro, “Pegasos: Primal estimated sub-gradient solver for SVM,” in *Proc. ICML*, 2007.
- [21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [22] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, 2013, pp. 3111–3119.
- [23] J. Godfrey, E. Holliman, and J. McDaniel, “SWITCHBOARD: Telephone speech corpus for research and development,” in *Proc. ICASSP*, 1992.
- [24] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, “The Kaldi speech recognition toolkit,” in *Proc. ASRU*, 2011.
- [25] Y. Zhang and B. Wallace, “A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1510.03820*, 2015.
- [26] F. Chollet, “Keras,” <https://github.com/fchollet/keras>, 2015.
- [27] Theano Development Team, “Theano: A Python framework for fast computation of mathematical expressions,” *arXiv e-prints*, vol. abs/1605.02688, May 2016.
- [28] M. D. Zeiler, “Adadelta: an adaptive learning rate method,” *arXiv preprint arXiv:1212.5701*, 2012.
- [29] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv*, 2012.
- [30] R. Řehůřek and P. Sojka, “Software Framework for Topic Modelling with Large Corpora,” in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 2010, pp. 45–50.
- [31] J. Trmal, G. Chen, D. Povey, S. Khudanpur, P. Ghahremani, X. Zhang, V. Manohar, C. Liu, A. Jansen, D. Klakow *et al.*, “A keyword search system using open source software,” in *Proc. SLT*, 2014.