Interspeech 2018
2-6 September 2018, Hyderabad, India

Joint Learning of Domain Classification and Out-of-Domain Detection with Dynamic Class Weighting for Satisficing False Acceptance Rates

Joo-Kyung Kim and Young-Bum Kim

Amazon Alexa

{jookyk, youngbum}@amazon.com

Abstract

In domain classification for spoken dialog systems, correct detection of out-of-domain (OOD) utterances is crucial because it reduces confusion and unnecessary interaction costs between users and the systems. Previous work usually utilizes OOD detectors that are trained separately from in-domain (IND) classifiers, and confidence thresholding for OOD detection given target evaluation scores. In this paper, we introduce a neural joint learning model for domain classification and OOD detection, where dynamic class weighting is used during the model training to *satisfice* a given OOD false acceptance rate (FAR) while maximizing the domain classification accuracy. Evaluating on two domain classification tasks for the utterances from a large spoken dialogue system, we show that our approach significantly improves the domain classification performance with satisficing given target FARs.

Index Terms: domain classification, out-of-domain detection, false acceptance rate, dynamic class weighting

1. Introduction

Domain classification is one of the three major components of spoken language understanding along with intent detection and slot filling [1]. Errors made by domain classifiers are more critical than errors by the other components because the domain classification errors tend to be propagated to completely incorrect system actions or responses. Since recent spoken dialog systems such as Amazon Alexa, Google Assistant, Microsoft Cortana, and Apple Siri deal with a wide variety of scenarios [2, 3], correct domain classification of a user's utterance into one of the supported domains or out-of-domain (OOD) is becoming more complex and important.

Domain classifiers are usually trained focusing on maximizing a single evaluation metric such as classification accuracy and F-score. In real spoken dialog systems, however, correctly detecting OOD utterances[1] is crucial because spoken dialog systems are prone to receive various OOD utterances such as unactionable utterances, ungrammatical utterances, and those with severe ASR errors. Therefore, misclassifying OOD utterances as in-domain (IND) causes confusion and unnecessary interaction costs between users and the dialog systems. To reduce the OOD misclassification of domain classifiers, false acceptance rate (FAR), which is 1 - OOD recall, is often regarded as a *satisficing* metric that must be below a predefined value while false rejection rate (FRR), which is 1 - IND recall, is considered relatively less important.

Previous approaches for OOD detection usually train OOD detectors separately or on top of IND classifiers [4, 5, 6, 7]. In these methods, OOD detectors are trained only to identify whether the given utterances are OOD or not regardless the classification of IND utterances. Also, the methods are evaluated either based on IND recall and OOD recall [5] or Equal Error Rate (ERR), where thresholding is used to match FAR and FRR to be the same [4, 6]. Consequently, they do not specifically focus on keeping FARs to be low.

Dealing with those issues, we introduce a joint learning model of IND classification and OOD detection. Joint learning models have been shown effective for various spoken language understanding tasks. For example, joint training of intent detection and slot-filling [8], joint training of all the three SLU components [9, 10], and joint training of multiple domains [11, 12, 13] have been shown synergistic since the jointly trained components are highly related to each other. Our model jointly trains a multi-class classifier for the domain classification and a binary classifier for the OOD detection on top of a bidirectional Long Short-Term Memory (BiLSTM) layer [14] for the utterance representations. Within this joint architecture, IND classification and OOD detection are helpful to each other by sharing underlying vector representations. In addition, we use dynamic class weighting, where we adjust the class weights for the IND and OOD loss functions to satisfice the FAR on the development set for each epoch. With dynamic class weighting, we first focus on the FAR as a satisficing metric that must be equal or lower than a predefined target value and then the IND accuracy as an optimizing metric.

Evaluating on two datasets collected by Amazon Alexa, we show that our joint learning model, which aims to satisfice the FAR and maximize the overall classification accuracy with dynamic class weighting, significantly improves domain classification performance given the two metrics.

2. Satisficing false acceptance rates

Our objective in this paper is having FAR as a satisficing metric and the domain classification accuracy as an optimizing metric. This is relevant to addressing class imbalance or unequal class cost cases in classification, where techniques such as oversampling, undersampling, SMOTE, class weighting, and threshold-moving are commonly used [15, 16, 17, 18, 19]. However, our objective is different from class imbalance or unequal class costs because of the following reasons:

- We have two metrics (FAR and accuracy) to optimize rather than one.
- FAR must be satisficed to be equal or below a predefined target value.
- Proper oversampling rates or class weights for satisficing a given FAR is difficult to be decided in advance since they would be substantially different for different datasets.

We formulate the objective as a non-differentiable con-

[1] The OOD explicitly acknowledges systems' inability to respond to the user's request whenever it does not have a valid response.

10.21437/Interspeech.2018-1581

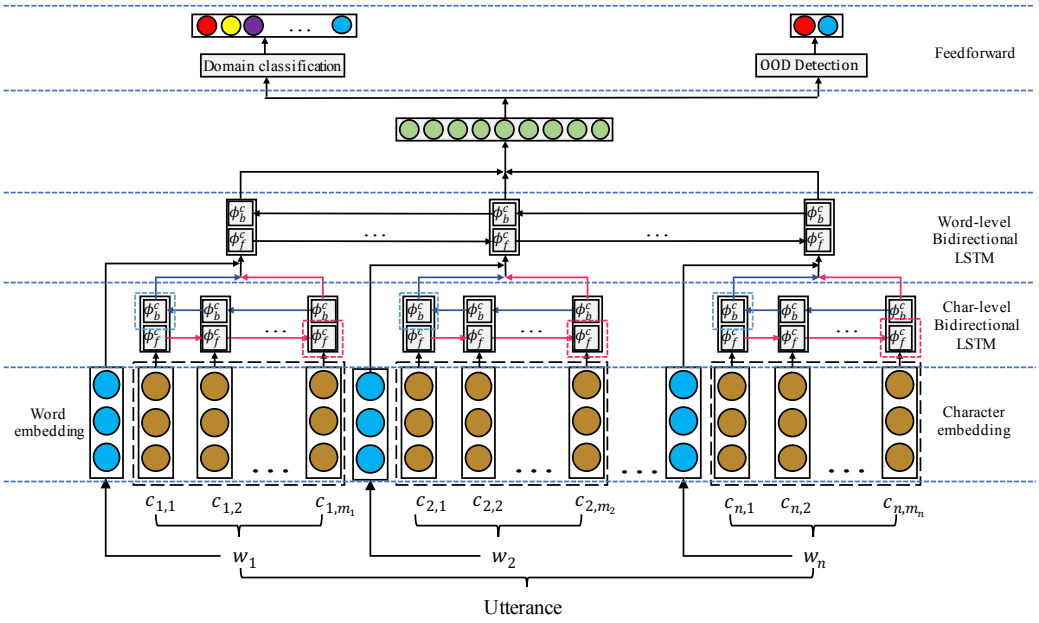

Figure 1: *Model architecture: each word is represented by the concatenation of the word vector from word embedding and the orthography-sensitive vector from the last outputs of two character LSTMs. On top of the word representations, we use a BiLSTM to represent the word sequence as a vector sequence. The last outputs of the two LSTMs are concatenated to be used as a single vector representing the entire utterance. On top of that, we jointly train a domain classifier and an OOD detector.*

strained optimization problem as follows:

$$\max accuracy\,(\mathcal{D}) \quad \text{subject to} \quad FAR\,(\mathcal{D}) \leq T, \quad (1)$$

where $\mathcal{D}$ is an evaluated dataset and T is a target FAR. In Section 3, we describe our model and formulate a differentiable surrogate loss function to address the objective.

3. Model

Figure 1 shows the overall architecture of the proposed joint model of domain classification and OOD detection.

3.1. Word representation

In order to leverage character-level vector representations, we use both character embeddings and word embeddings to represent each word [20]. Let $\mathcal{C}$ and $\mathcal{W}$ denote the set of characters and the set of words, respectively. Let $\oplus$ denote the vector concatenation operator. We formulate LSTM [21] as a function $\phi : \mathbb{R}^d \times \mathbb{R}^{d'} \rightarrow \mathbb{R}^{d'}$ that takes an input vector x and a state vector h to output a new state vector $h' = \phi(x, h)$.

The model parameters associated with the word representations are:

$$\textbf{Char embedding: } e_c \in \mathbb{R}^{25} \text{ for each } c \in \mathcal{C}$$
$$\textbf{Char LSTMs: } \phi_f^{\mathcal{C}}, \phi_b^{\mathcal{C}} : \mathbb{R}^{25} \times \mathbb{R}^{25} \rightarrow \mathbb{R}^{25}$$
$$\textbf{Word embedding: } e_w \in \mathbb{R}^{100} \text{ for each } w \in \mathcal{W}$$

Let $(w_1, \ldots, w_n)$ denote a word sequence where word w_i has a character $w_i(j) \in \mathcal{C}$ at position j. The vector representation of the i-th word, $v_i \in \mathbb{R}^{150}$, is obtained by the concatenation of the both ends of the character LSTM outputs and the i-th word vector e_{w_i}.[2] It is formulated as follows:

$$f_j^{\mathcal{C}} = \phi_f^{\mathcal{C}}\left(e_{w_i(j)}, f_{j-1}^{\mathcal{C}}\right) \qquad \forall j = 1 \ldots |w_i|$$
$$b_j^{\mathcal{C}} = \phi_b^{\mathcal{C}}\left(e_{w_i(j)}, b_{j+1}^{\mathcal{C}}\right) \qquad \forall j = |w_i| \ldots 1$$
$$v_i = f_{|w_i|}^{\mathcal{C}} \oplus b_1^{\mathcal{C}} \oplus e_{w_i} \qquad ,$$

where $f_{|w_i|}^{\mathcal{C}}$, $b_1^{\mathcal{C}}$ denote the forward LSTM output for the last character and the backward LSTM output for the first character, respectively.

3.2. Utterance representation

We encode the word vector sequence $(v_1, \ldots, v_n)$ with a BiLSTM for each $i = 1, \ldots, n$.[3]:

$$\textbf{Word LSTMs: } \phi_f^{\mathcal{W}}, \phi_b^{\mathcal{W}} : \mathbb{R}^{150} \times \mathbb{R}^{100} \rightarrow \mathbb{R}^{100}.$$

$$f_i^{\mathcal{W}} = \phi_f^{\mathcal{W}}\left(v_i, f_{i-1}^{\mathcal{W}}\right) \qquad \forall i = 1 \ldots n$$
$$b_i^{\mathcal{W}} = \phi_b^{\mathcal{W}}\left(v_i, b_{i+1}^{\mathcal{W}}\right) \qquad \forall i = n \ldots 1.$$

Then, in a similar way to obtaining the word representations from the character BiLSTMs, we concatenate the last outputs of both word LSTMs to represent the whole utterance as a single 200 dimensional vector:

$$u = f_n^{\mathcal{W}} \oplus b_1^{\mathcal{W}}.$$

We also evaluate two other utterance representation methods, which are word vector summation $u = \sum_{i=1}^{n} v_i$ and convolutional neural networks (CNN).[4]

[2] e_w is pretrained with GloVe leveraging Wikipedia 2014 and Gigaword 5 corpora [22].

[3] $f_0^{\mathcal{C}}, b_{|w_i|+1}^{\mathcal{C}}, f_0^{\mathcal{W}}$, and $b_{|n|+1}^{\mathcal{W}}$ are randomly initialized vectors.

[4] In a similar method to [23], we use three convolution filters whose

3.3. Domain classification

On top of the utterance vector u, we use a feed-forward neural network ϕ_d and softmax function to obtain the probability distribution over the entire domains[5] as:

$$d = softmax\left(\phi_d\left(u\right)\right).$$

The loss function for the domain prediction is formulated as the cross-entropy between the label and the probability distribution of an utterance:

$$\mathcal{L}_D\left(d\right) = -\hat{d}\log d, \tag{2}$$

where $\hat{d}$ is the one-hot representation of the ground-truth domain of the current utterance.

3.4. OOD detection

Along with the domain classifier, we jointly train an OOD detector, which predicts whether the current utterance is IND or OOD. We use a feed-forward network ϕ_o as did for the domain classification:

$$o = softmax\left(\phi_o\left(u\right)\right).$$

The loss function for OOD detection is formulated as:

$$\mathcal{L}_O\left(o\right) = -\hat{o}\log o, \tag{3}$$

where $\hat{o}$ is a two dimensional one-hot vector representing whether the current utterance is IND or OOD.

3.5. Joint loss function

A loss function for combining domain classification and OOD detection is formulated as follows:

$$\mathcal{L}_J\left(\cdot\right) = \mathcal{L}_D\left(\cdot\right) + \alpha\mathcal{L}_O\left(\cdot\right), \tag{4}$$

where α is a hyperparameter that controls the degree of the influence from the binary OOD detector. We show results on different α values in Section 4.2.

3.6. Dynamic class weighting

The final loss function, which approximately optimizes Equation 1, is as follows:

$$\mathcal{L}\left(\mathcal{D}\right) = (2-\lambda)\sum_{d_k\in\mathcal{D}_{IND}}\mathcal{L}_J\left(d_k\right) + \lambda\sum_{d_l\in\mathcal{D}_{OOD}}\mathcal{L}_J\left(d_l\right), \tag{5}$$

where $\mathcal{D}_{IND}$ is the set of utterances with IND ground-truths, $\mathcal{D}_{OOD}$ is the utterance set with OOD ground-truths, $\mathcal{L}_J$ is the joint loss function in Section 3.5, and λ is a parameter deciding the class weights for IND domains and the OOD. Here, $\sum_{d_k\in\mathcal{D}_{IND}}\mathcal{L}_J\left(d_k\right)$ and $\sum_{d_l\in\mathcal{D}_{OOD}}\mathcal{L}_J\left(d_l\right)$ are surrogate loss functions for maximizing the IND classification and satisficing the FAR, respectively. This formulation uses $2-\lambda$ and λ as the class weights for IND and OOD, respectively. The main issue of Equation 5 is that we cannot predetermine λ as aforementioned.

To obtain a proper λ, we introduce dynamic class weighting for OOD, where λ is changed during the training so that the FAR on the development set is satisficed with minimal λ increase for OOD. We initialize λ to 1 so that the class weights for both IND and OOD are 1 in the beginning. At the end of each training epoch, we calculate the FAR on the development set. If the current FAR does not satisfice the target FAR, we add γ to current λ. Oppositely, if the target FAR is satisficed, we subtract γ from current λ. When adding and subtracting γ, we limit the λ value to be less than 2 and larger than 0. In our work, we initialize γ to 0.1.[6] To reduce fluctuations of λ during the late epochs, we halve γ each time when the target FAR is satisficed in the current epoch but not in the previous epoch. With this approach, we encourage the model to find the minimal class weight change that satisfices the FAR and then focuses on the overall classification accuracy.

sizes are 3, 4, and 5 on top of a word vector sequence. Then, we use max pooling for each filter output, and finally concatenate the three max pooling outputs to represent the whole utterance.

[5]We use a single hidden layer with SeLU activation function [24] for normalized activation outputs.

	21 domain dataset			1,500 skill dataset		
	IND	OOD	Total	IND	OOD	Total
Train	712k	255k	967k	372k	381k	753k
Dev	112k	17k	129k	103k	35k	138k
Test	112k	21k	133k	105k	35k	104k

Table 1: *The numbers of the utterances in the two datasets: 21 Alexa domains and 1,500 Alexa skills.*

4. Experiments

We have conducted a series of experiments to evaluate the proposed method on datasets obtained from real usage data in Amazon Alexa with two domain classification tasks.

4.1. Datasets

We evaluate our models on two domain classification tasks from different data sources: (1) utterances from 21 Alexa domains, (2) utterances from frequently used 1,500 skills out of more than 40,000 skills.[7] For both cases, we use randomly sampled unique utterances that are collected and annotated from the real user logs. The average utterance lengths are 5.96 and 5.68 for the 21 domains and the 1,500 skills, respectively. Table 1 shows the statistic of the datasets.

4.2. Results

Each evaluated model is trained for 50 epochs and we use the parameters at the epoch showing the best score on the development set to report the scores on the test set. We use ADAM [27] with learning rate 0.001 for the optimization. For stable training, we use gradient clipping, where the threshold is set to 5. For efficiency, we use a variant of LSTM, where the input gate and the forget gate are coupled and peephole connections are used [28, 29]. For the LSTM regularization, we use variational dropout [30]. All the models are implemented with DyNet [31].

Table 2 and 3 show the classification accuracies on the two datasets with various models given different target FARs. Even though the FAR on the development set is satisficed with dynamic class weighting, the FAR on the test set might not be satisficed. To satisfice each given target FAR for the test set, we set a decision threshold to regard a predicted domain as OOD when the highest confidence score is below the threshold.

In Table 2 and 3, *Separate* models use separate underlying utterance representations for IND only classification and OOD

[6]We also tried different values but there were no significant differences in the experiment results.

[7]In Amazon Alexa, a *skill* is a domain developed by third-party developers [25, 26].

Model	α	Target FAR		
		6%	5.5%	5%
Separate (BiLSTM)	1	85.69	84.68	83.7
Joint (WordVecSum)	0	87.28	86.85	86.37
Joint (CNN)	0	88.61	88.18	87.83
Joint (BiLSTM)	0	89.2	88.73	88.27
	0.001	89.33	88.94	88.4
	0.005	89.25	88.84	88.36
	0.01	89.27	88.89	88.38
	0.05	89.26	88.88	88.43
Joint (BiLSTM) w/ dynamic class weighting	0	90.67	90.52	90.26
	0.001	90.67	90.52	90.34
	0.005	**90.71**	90.53	**90.38**
	0.01	90.69	**90.61**	90.28
	0.05	90.63	90.53	90.30

Table 2: *The test classification accuracies (%) of various models given different satisficing FARs on the 21 domain dataset. α of Equation 4 is set to 0 for Separate case and 1 for all the other cases. α is the coefficient for the OOD detector loss.*

Model	α	Target FAR		
		2%	1.5%	1%
Separate (BiLSTM)	1	77.50	75.40	72.41
Joint (WordVecSum)	0	74.07	72.31	69.28
Joint (CNN)	0	77.69	75.95	73.50
Joint (BiLSTM)	0	78.19	76.48	74.10
	0.001	78.37	76.97	74.32
	0.005	78.14	76.59	74.25
	0.01	78.32	76.44	74.34
	0.05	78.05	76.36	73,97
Joint (BiLSTM) w/ dynamic class weighting	0	79.18	78.26	76.74
	0.001	79.26	78.63	77.22
	0.005	**79.34**	**78.91**	**77.32**
	0.01	79.20	78.54	77.05
	0.05	79.13	78.27	77.07

Table 3: *The test classification accuracies (%) on the 1,500 skill dataset.*

detection. In this case, given an utterance, we first run the OOD detector to predict whether the utterance belongs to IND or OOD. If it is predicted as IND, we run the IND classifier to predict the domain of the utterance.[8]

Joint models share the underlying utterance representations for both domain classification and OOD detection. The domain classifier also includes OOD as one of the domains so that the domain classifier can also learn representations from OOD utterances. As aforementioned in Section 3.2, we also evaluate the other utterance representation methods, word vector summation and CNN. We utilize the OOD detector with setting α, which is the coefficient for the OOD detection loss in Equation 4.

Joint with dynamic class weighting models are trained including dynamic class weighting with given target FARs for the development sets.

4.2.1. 21 Alexa Domains

This task classifies input utterances to either one of 21 Alexa domains or OOD. For example, the domains for "*What's the weather this weekend in Orlando*," "*Get me a ride to Seattle airport*," and "*Oh no nothing*" should be classified as `Weather`, `BookingAndReservations`, and `OOD`, respectively.

When no target FAR is given, the accuracy and the FAR of the *Joint (BiLSTM)* model with $\alpha = 0$ for this dataset are 91.06% and 9.75%, respectively.

We evaluate the proposed models with 6%, 5.5%, and 5% as the target FARs. Table 2 shows the model evaluation results.

Since domain classification and OOD detection are closely related tasks, it is shown that *Joint (BiLSTM)* models outperform *Separate (BiLSTM)* model for all the cases. Also, to represent the utterances in a vector space, using BiLSTM is shown to be consistently better than using word vector summation and CNN in our experiments.

For *Joint* models, utilizing the OOD detector by setting α to be higher than 0 during the training shows better accuracies than not using it. This demonstrates that jointly training a separate OOD detector noticeably helps increase the overall classification performance.

We can observe that the accuracies of *Joint with dynamic class weighting* models are significantly higher than those of the other models. This shows that utilizing dynamic class weighting is effective for our objective, where we first satisfice a given FAR and then maximize the accuracy by dynamically finding more effective class weights.

4.2.2. 1,500 Alexa Skills

This task deals with utterance classification to either one of 1,500 skills or OOD. For example, the skills for "*what does a peacock say*" and "*find me the recipe for world's best lasagna*" should be predicted as `ZooKeeper` and `AllRecipes`, respectively. The skills are significantly more diverse and less well defined than 21 Alexa domains, which makes the classification more challenging. In real spoken dialog systems, the classification performance can be further improved by leveraging various contextual information [32, 33, 34, 26]. However, they are beyond the scope of this paper, and we leave the evaluation of our models on such reranking systems as future work.

On this task, when there is no target FAR, the accuracy and the FAR of the *Joint (BiLSTM)* model with $\alpha = 0$ are 80.65% and 3.62%, respectively.

Therefore, we have evaluated our models on lower target FARs, 2%, 1.5%, and 1%. Table 3 shows the results of our proposed models. Overall, similarly to the results of 21 Alexa Domains, we can see that *Joint* models are better than *Separate* model, using BiLSTM outperforms using word vector summation or CNN for the utterance representations, and *Joint with dynamic class weighting* models show significantly better performance than other models.

5. Conclusion

We have introduced a joint learning model of domain classification and OOD detection utilizing dynamic class weighting to satisfice a target FAR and then maximize the overall classification accuracy. Evaluating on two domain classification tasks for the utterances from Amazon Alexa, we have shown that our proposed joint learning models with dynamic class weighting is more effective than the models with separate learning of domain classification and OOD detection or those trained to optimize a single metric when we have FAR as a satisficing metric and accuracy as an optimizing metric.

6. References

[1] G. Tur and R. de Mori, *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. New York,

[8] Since the OOD detector is solely trained in *Separate* models, we do not need to set α to be relatively low.

NY: John Wiley and Sons, 2011.

[2] R. Sarikaya, P. A. Crook, A. Marin, M. Jeong, J.-P. Robichaud, A. Celikyilmaz, Y.-B. Kim, A. Rochette, O. Z. Khan, X. Liu *et al.*, "An overview of end-to-end language understanding and dialog management for personal digital assistants," in *IEEE Workshop on Spoken Language Technology (SLT)*, 2016, pp. 391–397.

[3] R. Sarikaya, "The technology behind personal digital assistants: An overview of the system architecture and key components," *IEEE Signal Processing Magazine*, vol. 34, no. 1, pp. 67–81, 2017.

[4] I. Lane, T. Kawahara, and T. Matsui, "Out-of-domain utterance detection using classification confidences of multiple topics," *Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 15, no. 1, pp. 150–161, 2007.

[5] G. Tur, A. Deoras, and D. Hakkani-Tür, "Detecting out-of-domain utterances addressed to a virtual personal assistant," in *Interspeech*, 2014, pp. 283–287.

[6] S. Ryu, S. Kim, J. Choi, H. Yu, and G. G. Lee, "Neural sentence embedding using only in-domain sentences for out-of-domain sentence detection in dialog systems," *Pattern Recognition Letters*, vol. 88, pp. 26–32, 2017.

[7] K.-J. Oh, D. Lee, C. Park, H.-J. Choi, Y.-S. Jeong, S. Hong, and S. Kwon, "Out-of-domain detection method based on sentence distance for dialogue systems," in *IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2018, pp. 673–676.

[8] J.-K. Kim, G. Tur, A. Celikyilmaz, B. Cao, and Y.-Y. Wang, "Enriching word embedding for intent detection," in *IEEE Workshop on Spoken Language Technology (SLT)*, 2016, pp. 414–419.

[9] D. Hakkani-Tür, G. Tur, A. Celikyilmaz, Y.-N. Chen, J. Gao, L. Deng, and Y.-Y. Wang, "Multi-Domain Joint Semantic Frame Parsing using Bi-directional RNN-LSTM," in *Interspeech*, 2016.

[10] Y.-B. Kim, S. Lee, and K. Stratos, "OneNet: Joint domain, intent, slot prediction for spoken language understanding," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017.

[11] Y.-B. Kim, K. Stratos, R. Sarikaya, and M. Jeong, "New transfer learning techniques for disparate label sets," in *Proceedings of annual meeting of the association for computational linguistics (ACL)*, 2015, pp. 473–482.

[12] A. Jaech, L. Heck, and M. Ostendorf, "Domain adaptation of recurrent neural networks for natural language understanding," in *Interspeech*, 2016.

[13] Y.-B. Kim, K. Stratos, and R. Sarikaya, "Frustratingly easy neural domain adaptation," in *COLING*, 2016, pp. 387–396.

[14] A. Graves and J. Schmidhuber, "Frame-wise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005.

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[16] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, 2002.

[17] M. A. Maloof, "Learning when data sets are imbalanced and when costs are unequal and unknown," in *ICML-2003 workshop on learning from imbalanced data sets*, 2003.

[18] Z.-H. Zhou and X.-Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 18, no. 1, pp. 63–77, 2006.

[19] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," in *arXiv:1710.05381*, 2017.

[20] B. Plank, A. Søgaard, and Y. Goldberg, "Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss," in *Proceedings of annual meeting of the association for computational linguistics (ACL)*, 2016, pp. 412–418.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

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

[23] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1292–1302.

[24] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Advances in Neural Information Processing Systems 30 (NIPS)*, 2017, pp. 972–981.

[25] A. Kumar, A. Gupta, J. Chan, S. Tucker, B. Hoffmeister, M. Dreyer, S. Peshterliev, A. Gandhe, D. Filiminov, A. Rastrow, C. Monson, and A. Kumar, "Just ASK: Building an Architecture for Extensible Self-Service Spoken Language Understanding," in *NIPS Workshop on Conversational AI*, 2017.

[26] Y.-B. Kim, D. Kim, A. Kumar, and R. Sarikaya, "Efficient Large-Scale Neural Domain Classification with Personalized Attention," in *ACL*, 2018.

[27] D. P. Kingma and J. L. Ba, "ADAM: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[28] F. A. Gers and J. Schmidhuber, "Recurrent Nets that Time and Count," in *IJCNN*, vol. 3, 2000, pp. 189–194.

[29] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *Transactions on Neural Network Learning and Systems (TNNLS)*, vol. 28, no. 10, pp. 2222–2232, 2017.

[30] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in Neural Information Processing Systems 29 (NIPS)*, 2016, pp. 1019–1027.

[31] G. Neubig, C. Dyer, Y. Goldberg, A. Matthews, W. Ammar, A. Anastasopoulos, M. Ballesteros, D. Chiang, D. Clothiaux, T. Cohn *et al.*, "DyNet: The Dynamic Neural Network Toolkit," *arXiv preprint arXiv:1701.03980*, 2017.

[32] J.-P. Robichaud, P. A. Crook, P. Xu, O. Z. Khan, and R. Sarikaya, "Hypotheses ranking for robust domain classification and tracking in dialogue systems," in *Interspeech*, 2014, pp. 145–149.

[33] P. A. Crook, J.-P. Martin, and R. Sarikaya, "Multi-language hypotheses ranking and domain tracking for open domain," in *Interspeech*, 2015.

[34] Y.-B. Kim, D. Kim, J.-K. Kim, and R. Sarikaya, "A scalable neural shortlisting-reranking approach for large-scale domain classification in natural language understanding," in *NAACL*, 2018, pp. 16–24.