INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Paraphrase Label Alignment for Voice Application Retrieval in Spoken Language Understanding

Zheng Gao, Radhika Arava, Qian Hu, Xibin Gao, Thahir Mohamed, Wei Xiao, Mohamed AbdelHady

Amazon Alexa AI, Seattle, USA

{zhenggao, aravar, huqia, gxibin, thahirm, weixiaow, mbdeamz}@amazon.com

Abstract

Spoken language understanding (SLU) smart assistants such as Amazon Alexa host hundreds of thousands of voice applications (skills) to delight end-users and fulfill their utterance requests. Sometimes utterances fail to be claimed by smart assistants due to system problems such as model incapability or routing errors. The failure may lead to customer frustration, dialog termination and eventually cause customer churn. To avoid this, we design a skill retrieval system as a downstream service to suggest fallback skills to unclaimed utterances. If the suggested skill satisfies customer intent, the conversation will be recovered with the assistant. For the sake of smooth customer experience, we only present the most relevant skill to customers, resulting in partial observation problem which constrains retrieval model training. To solve this problem, we propose a two-step approach to automatically align claimed utterance labels to unclaimed utterances. Extensive experiments on two real-world datasets demonstrate that our proposed model significantly outperforms a number of strong alternatives.

Index Terms: spoken language understanding, data augmentation, voice application retrieval, label alignment

1. Introduction

Spoken language understanding (SLU) systems play an essential role in smart assistants such as Amazon Alexa and Google Assistant by invoking the most appropriate voice applications (aka skills) to respond to customer utterances. Figure 1 shows a standard SLU system consisting of two components: automatic speech recognition (ASR) and natural language understanding (NLU) [1]. ASR component first converts the audio signal of an utterance into text, NLU component thereafter claims an appropriate skill for response. Specifically, NLU component provides three modules including domain classifier (DC), intent classifier (IC) and named entity recognition (NER). For example, after an utterance "*play today's hits*" is recognized by ASR component, NLU component further processes it and finally launches "*Pandora*" to play trending music in the smart assistant.

Sometimes utterances fail to be claimed by NLU component due to system problems such as model incapability or routing errors. To reduce customer friction and recover the conversation, we propose a downstream service (highlighted in Figure 1) to suggest fallback skills to unclaimed utterances. And a skill will be launched if customer accepts the suggestion. For the sake of smooth customer experience, we only retrieve the most relevant skill to customers, resulting in partial observation problem where only the retrieved skill can receive its explicit customer feedback ("*positive*" or "*negative*") while rest of skills' relevancy remains unknown.

To solve the label scarcity challenge, there are two major research tracks from either interior or exterior perspective. Approaches from interior perspective aim to uncover hidden labels within original data via self-training or positive-unlabeled learning techniques. Self-training approaches [2–5] are semi-supervised models trained on labeled data to guide soft label matching on correlated unlabeled data. Positive-unlabeled learning approaches [6–8] iteratively train models to predict on unlabeled data and bring in positive pseudo labels for augmented training. Approaches from exterior perspective exploit external labeled data via cross-domain or knowledge distillation techniques. Cross-domain approaches [9–13] learn domain mapping functions to adequately propagate source domain information to target domain. Knowledge distillation approaches [14–17] utilize teacher models trained on external labeled data to guide student model optimization on the original data.

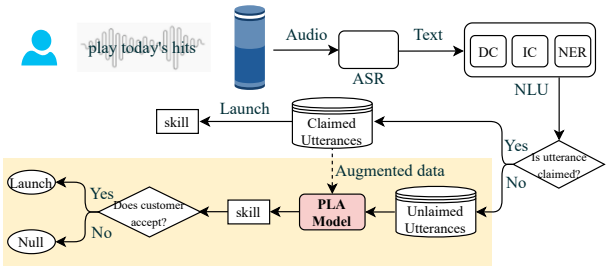


Figure 1: *The use case of PLA model in SLU systems.*

However, all aforementioned approaches are classification oriented models to predict labels on all classes. They may not work with hundreds of thousands of developed skills. To mitigate the label scarcity effect in unclaimed utterances' skill retrieval, we propose an Paraphrase Label Alignment (PLA) model by taking advantage of claimed utterances and their launched skills. It is a two-step approach to first detect paraphrases from claimed utterances and then retrieve skills via a customized loss function aligning paraphrase launched skills with target unclaimed utterances. In the end, skills predicted with the highest relevancy scores are suggested to unclaimed utterances. We assume that an unclaimed utterance is a perturbation from a set of claimed paraphrases. The label alignment is a type of skill label projection in order to increase model robustness against the noise introduced in unclaimed utterances.

2. Method

2.1. Problem Formulation

Given unclaimed utterance set U, each utterance $u \in U$ has either "*positive*" or "*negative*" label to the suggested skill from explicit customer feedback. The rest of skills are all labeled as "*unknown*". Similarly, given claimed utterance set P, each utterance $p \in P$ has "*positive*" label to its launched skill and "*unknown*" labels to the rest of skills because empirically launched skills of SLU systems are relevant to utterances with high confidence. PLA model utilizes both claimed and unclaimed ut-

http://dx.doi.org/10.21437/Interspeech.2021-97

terances to mitigate disruptive and scarce label problem. Algorithm 1 illustrates the overall training procedure including a paraphrase detection step (Section 2.2) which trains a Siamese network model $\mathcal{S}$ to select top relevant paraphrases and a skill retrieval step (Section 2.3) which trains a Reranker model $\mathcal{R}$ to predict skill relevancy. In the testing stage, we only apply the skill retrieval step on unclaimed utterances to predict relevancy labels of their Shortlisting skills and retrieve the skill with the highest prediction score in the end. Claimed utterances and the paraphrase detection step are not involved in the testing stage.

Algorithm 1 PLA Model Training

Input: unclaimed utterance u, claimed utterance set P, raw selection number N, pairwise matching threshold ρ, Shortlisting skill length E, paraphrase number K;
Output: Siamese network model $\mathcal{S}$, Reranker $\mathcal{R}$;

P_s = ParaphraseDetection(u, P, N, ρ);
SkillRetrieval(u, P_s, E, K);

ParaphraseDetection(u, P, N, ρ):
  For utterance u, compute Top N paraphrases $P_r \subseteq P$ with highest score sum on Jaccard and Faiss similarity;
  Train Siamese network model $\mathcal{S}$ using claimed data P;
  Predict all pairwise labels between u and P_r;
return $P_s \subseteq P_r$ with similarity score $\geq \rho$;

SkillRetrieval(u, P_s, E, K):
  Retrieve top E relevant skills V for u via TF-IDF;
  Select Top K paraphrases $P_k \subseteq P_s$ having skills $\in V$;
  Train Reranker $\mathcal{R}$ based on u and P_k;

2.2. Paraphrase Detection

2.2.1. Raw Selection

A deterministic model is designed to select paraphrase candidates based on lexical and semantic similarity without training effort. From lexical perspective, Jaccard Similarity $\mathcal{J}(\cdot)$ computes the word level similarity between unclaimed utterance u and each claimed utterance $p \in P$. From semantic perspective, the hidden state of class token in pretrained BERT model [18] is used to represent sentence vectors of u and p. And their semantic similarity is calculated by Faiss model $\mathcal{F}(\cdot)$ [19].

$$sim(u,p) = \mathcal{J}(u,p) + \mathcal{F}(\text{BERT}(u), \text{BERT}(p)) \quad (1)$$

$sim(u,p)$ represents the summed similarity score. In the end, for utterance u, the top N utterances $P_r \subseteq P$ with the highest summed similarity scores are selected as paraphrase candidates.

2.2.2. Pairwise Matching

The raw selection process detects task-independent paraphrases solely based on utterance content matching. In this section, we trains a Siamese network model $\mathcal{S}$ to predict the task-dependent relationships between utterance u and paraphrase candidates P_r. The training data is constructed from claimed utterances P as their labels are more trustable. Derived from Sentence-BERT [20], the model inputs are utterance pairs and outputs are binary labels indicating whether two utterances are launched with the same skill or not. Given an utterance pair $\{p_1, p_2\} \in P$ and its associated binary label $l \in \{0, 1\}$, we encode their sentences and minimize their contrastive loss $\mathcal{L}_{pd}$ [21]:

$$\begin{aligned} \mathcal{D}_p &= W_p | \sum_{w \in p_1} \text{encoder}(w) - \sum_{w \in p_2} \text{encoder}(w) | \\ \mathcal{L}_{pd} &= l \cdot \mathcal{D}_p^2 + (1-l) \cdot \max(0, m - \mathcal{D}_p)^2 \end{aligned} \quad (2)$$

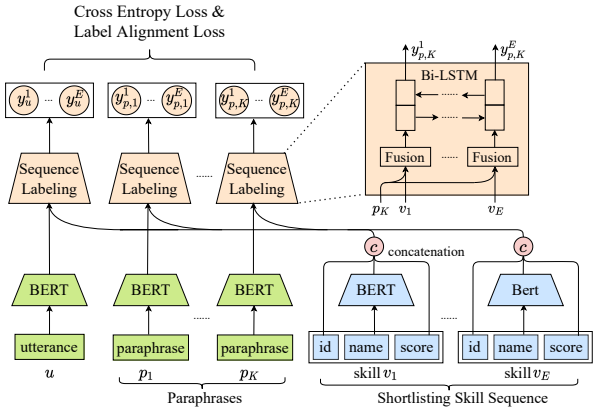


Figure 2: *Reranker model $\mathcal{R}$ with Paraphrase label alignment.*

where W_p is the weight matrix to calculate utterance pairwise distance $\mathcal{D}_p$. m is the margin to tighten the constraint. If two utterances are not launched with the same skill, their pairwise distance should be at least m, otherwise a loss will be occurred.

After the Siamese network model $\mathcal{S}$ is trained, for utterance u, we predict the binary labels for all pairs between u and $p \in P_r$. Only paraphrase candidates with matching score above the threshold ρ are retained to form final paraphrase set P_s.

2.3. Skill Retrieval

A two-step approach is proposed for skill retrieval with Shortlisting and Reranking. Shortlisting step utilizes lexical keyword matching to filter out most of irrelevant skills to utterances, so that the subsequent Reranking step only needs to predict skill relevancy labels on a truncated skill subset. Moreover, to best retain the ranking dependencies of Shortlisting skill sequences, a listwise Reranker model $\mathcal{R}$ (in Figure 2) is proposed to predict relevancy label sequences for the Shortlisting skill sequences.

2.3.1. Shortlisting

Shortlisting retrieves the top E most relevant skills of utterances from all skill candidates. In this paper, TF-IDF [22] algorithm is employed to efficiently calculate similarity score between utterance u and each skill v as:

$$score(u,v) = \sum_{w \in u} \frac{N_{wv}}{|v|} \log \frac{|S|}{|\{j : w \in S_j\}|} \quad (3)$$

For each word w in utterance u, N_{wv} denotes the word count of w in skill description of v; $|v|$ denotes skill description length; $|S|$ denotes the total number of skills; $|\{j : w \in S_j\}|$ denotes the number of skills with descriptions containing word w. In the end, we calculate all skill pairwise similarity scores for utterance u, and retrieve the Top E most relevant skills.

2.3.2. Reranking

Reranker model $\mathcal{R}$ aims to predict Shortlisting skill label sequences for both utterance u and its detected paraphrases. To insist a high standard on paraphrases, only the Top K paraphrases $P_k = \{p_1, ..., p_K\}$ having launched skills in the top E relevant skills $V = \{v_1, ..., v_E\}$ are kept for model training.

2.3.2.1 Utterance and Skill Encoder

Three types of information are considered to construct profile of each skill $v \in V$, including skill id, skill name and

skill Shortlisting score bin. Byte-pair-encoding (BPE) [23] and BERT [18] are used to encode sentences.

$$\begin{aligned} h_u &= \text{BERT}(\text{bpe}(u), \text{pos}(u)) \\ h_p &= \text{BERT}(\text{bpe}(p), \text{pos}(p)) \\ h_{na} &= \text{BERT}(\text{bpe}(v_{na}), \text{pos}(v_{na})) \\ h_s &= [v_{id}; h_{na}; v_{sc}] \end{aligned} \tag{4}$$

h_u, h_p and h_{na} denote the vector of utterance u, vector of paraphrase $p \in P_k$, and vector of skill name v_{na} respectively. And the concatenation of skill id embedding v_{id}, skill name vector h_{na} and skill score bin embedding v_{sc} form the integrated vector representation h_s of skill v. The fine tuned BERT parameters are shared across all three inputs $\{u, p, v_{na}\}$.

2.3.2.2 Sequence Labeling

This step predicts utterances/paraphrases label sequences on the Shortlisting skill sequences. Given utterance u and skill v, a Fusion layer is proposed to measure their relevance with two interaction features including element-wise product and absolute difference. To enable such calculation, we map u and v to the same latent space using one layer transformation. And the two interaction features are concatenated with utterance/skill original vectors as their pairwise relationship representation.

$$\begin{aligned} \hat{h}_u &= \text{ReLU}(W_u h_u + b_u) \\ \hat{h}_s &= \text{ReLU}(W_s h_s + b_s) \\ e_u &= \text{dropout}([\hat{h}_u; \hat{h}_s; \hat{h}_u \cdot \hat{h}_s; |\hat{h}_u - \hat{h}_s|]) \end{aligned} \tag{5}$$

Ws and bs denote related weights and bias, ReLU$(\cdot)$ is the applied activation function. In the end, the Fusion layer learns an interaction vector e_u for utterance u and skill v as the input of each decoding step with a dropout mechanism. The relevance between paraphrase p and skill v is calculated in the same way.

For utterance u and Shortlisting skill sequence V, $E_u = \{e_u^1, ..., e_u^E\}$ are calculated from Eq. 5 to predict label sequence $\hat{Y}_u = \{\hat{y}_u^1, ..., \hat{y}_u^E\}$. Bi-LSTM mechanism is employed to capture ranking skill dependencies. The i_{th} step output of Bi-LSTM o_u^i is used to predict the associated skill label $\hat{y}_u^i$.

$$\begin{aligned} o_u^i &= [\overrightarrow{o}_u^i; \overleftarrow{o}_u^i] \\ &= \text{Bi-LSTM}(e_u^i, \overrightarrow{o}_u^{i-1}, \overleftarrow{o}_u^{i+1}) \\ \hat{y}_u^i &= \sigma(W_o o_u^i + b_o) \end{aligned} \tag{6}$$

where o_u^i is the concatenation of the two directional LSTM outputs. Similarly, for the j_{th} paraphrase p_j, we can obtain its predicted label on the i_{th} skill as $\hat{y}_{p,j}^i$ with shared parameters.

2.3.2.3 Loss Function

Cross entropy loss $\mathcal{L}_{ce}$ sums over all cross entropy losses CE$(\cdot, \cdot)$ for each utterance/paraphrase prediction on each Shortlisting skill. $Y_u = \{y_u^1, ..., y_u^E\}$ is ground truth label sequence where y_u^i is the label of utterance u to the i_{th} skill. $y_u^i = 1$ if the label is "*positive*". $y_u^i = 0$ if the label is "*negative*" or "*unknown*". Paraphrase ground truth labels (i.e. $y_{p,j}^i$ of the j_{th} paraphrase on the i_{th} skill) are constructed in the same way.

$$\mathcal{L}_{ce} = \sum_i^E \text{CE}(y_u^i, \hat{y}_u^i) + \sum_i^E \sum_j^K \text{CE}(y_{p,j}^i, \hat{y}_{p,j}^i) \tag{7}$$

As we assume that unclaimed utterance u is a perturbation from its claimed paraphrases P_k, they should follow a similar label distribution on the same skill [24]. Therefore, a label alignment loss is proposed for each pair of utterance u and paraphrase $p \in P_k$. We minimize the KL divergence $\mathcal{D}_{KL}(\cdot||\cdot)$ of their predictions on the skills with ground truth label "*positive*" to paraphrase p but not "*positive*" to utterance u. It can be regarded as pseudo labeling to utterance u where $\mathbb{I}(y_p^i = 1, y_u^i = 0)$ is a binary indicator to select eligible skills.

$$\mathcal{L}_{kl} = \sum_p^{P_k} \sum_i^E \mathcal{D}_{KL}(\hat{y}_p^i || \hat{y}_u^i) \cdot \mathbb{I}(y_p^i = 1, y_u^i = 0) \tag{8}$$

In the end, the overall loss $\mathcal{L}$ is the weighted sum of average cross entropy loss and average label alignment loss:

$$\mathcal{L} = \frac{1}{E(1+K)} \mathcal{L}_{ce} + \frac{\alpha}{EK} \cdot \mathcal{L}_{kl} \tag{9}$$

3. Experiments

3.1. Dataset

Three real-world datasets are constructed from sampled one-month Alexa live traffic. Dataset en-US and en-CA are unclaimed English utterances from devices in the United States and Canada respectively. The third dataset is claimed English utterances from devices in both countries. In dataset en-US and en-CA, each utterance has a suggested skill and corresponding customer feedback. In the third dataset, each utterance has a successfully launched skill. In the end, we collect 1 million unclaimed utterances in en-US, 80 thousand unclaimed utterances in en-CA, and 16 million claimed utterances in both countries.

We construct two sets of data for paraphrase detection and skill retrieval. **First**, for paraphrase detection, all involved data comes from the claimed dataset. We randomly select 1 million utterance pairs with balanced labels for training and testing. **Second**, for skill retrieval, in both en-US and en-CA dataset, 20% of utterances with positive labels are evenly split for validation and testing. The remaining utterances are used for training. Claimed dataset are used to provide paraphrases.

3.2. Paraphrase Detection Evaluation

3.2.1. Encoder Structure Exploration

We evaluate several encoder structures of Siamese network model described in Section 2.2.2, including **TinyBERT** [25], **BERT** [18], **AlexaBERT** (BERT model fine tuned on Alexa data), **CharCNN** [26], **GloVe** [27], and **ELMo** [28]. For the first three BERT-based encoders, each utterance is represented by the hidden states of class token. For the second three word embedding models, each utterance is represented by the sum of word embeddings. All encoders are fine tuned on pretrained models and reported in Table 1. Precision, recall, and F1 score are reported as evaluation metrics.

3.2.2. Comparison Results

In table 1, we report the relative change of each baseline score $s(b)$ to our model score $s(a)$, calculated as $\frac{s(b)-s(a)}{s(a)}$.

We finally adopt ELMo as Siamese network sentence encoder as it achieves the best performance in Table 1. "-" means the relative performance degradation of each model to the best model ELMo. ELMo outperforms all BERT based models, indicating that Bi-LSTM structure can better capture utterance information under this task. GloVe performs the second best while CharCNN performs the worst, revealing that word level

Table 1: *Evaluation results for paraphrase detection. It represents the relative changes of all models to the best model ELMo.*

Model	Precision	Recall	F1
TinyBERT	-11.15%	-18.70%	-15.13%
Bert	-8.57%	-16.51%	-12.76%
AlexaBERT	-7.10%	-14.07%	-10.75%
GloVe	-6.46%	-8.74%	-7.62%
CharCNN	-16.95%	-21.32%	-19.20%
ELMo	**0.00%**	**0.00%**	**0.00%**

encoder works better than char level encoder. AlexaBERT performs better than default pretrained BERT due to different training data impact. While TinyBERT performs worse than BERT because of the trade-off between model size and performance.

3.3. Skill Retrieval Evaluation

3.3.1. Model Variants

As our model is composed of multiple components, many model variants can be derived from the original PLA model. To explore the best model structure, we compare our PLA model with the following truncated variants. 1) **TF-IDF**: PLA model Shortlisting step retrieving Top 1 skill with the highest TF-IDF score. 2) **Listwise**: PLA model Shortlisting and Reranking step trained on only unclaimed utterances without paraphrases. 3) **Listwise+PL**: Adding paraphrase labels (PL) directly to unclaimed utterances to train Listwise model. 3) **Listwise+Co**: Listwise model trained on the combination (Co) of unclaimed and claimed utterances. 4) **Faiss+JS+SR**: PLA model with only Faiss model and Jaccard similarity for paraphrase detection and skill retrieval (SR) step. 5)**Siamese+SR**: PLA model with only Siamese network for paraphrase detection and skill retrieval (SR) step. The first four variants ignore paraphrase information in model structure. While the last two variants are with truncated paraphrase detection step.

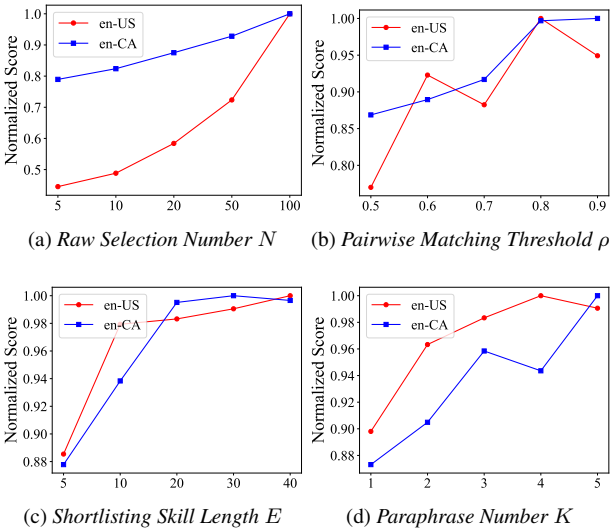
Skills with predicted relevancy score above 0.5 are ranked and retrieved. Their precision, recall, F1, and NDCG metrics are calculated and reported for model performance evaluation.

3.3.2. Comparison Results

Table 2: *Evaluation results for skill retrieval. It represents the relative changes of all model variants to the PLA model.*

Dataset	Model	Precision	Recall	F1	NDCG
en-US	TF-IDF	-45.17%	-56.69%	-54.21%	-54.57%
	Listwise	-54.05%	-61.45%	-60.89%	-59.65%
	Listwise+PL	-66.32%	**+3.49%**	-28.43%	+26.74%
	Listwise+Co	-12.58%	-5.38%	-9.81%	-9.56%
	Faiss+JS+SR	-9.98%	**+0.12%**	-8.64%	-9.22%
	Siamese+SR	-46.82%	-34.35%	-44.14%	-44.05%
en-CA	TF-IDF	-33.61%	-41.94%	-38.20%	-39.33%
	Listwise	-19.97%	-24.37%	-24.16%	-21.40%
	Listwise+PL	-30.25%	**+6.83%**	-14.12%	-15.24%
	Listwise+Co	**+1.00%**	-4.36%	-2.45%	-3.49%
	Faiss+JS+SR	**+11.66%**	-7.95%	-3.41%	-4.24%
	Siamese+SR	-22.16%	-29.49%	-26.90%	-26.81%

In Table 2, there are only few bold positive signs "+" showing that the corresponding model variant outperforms PLA model. In the rest cases, PLA model always achieves better performance than model variants. TF-IDF has the worst performance in both datasets to show its incapability in this task. Listwise+PL model indicates that simply incorporating paraphrase labels to utterances sacrifices model precision dramatically. Comparison results among Listwise, Listwise+Co and PLA model reveal that incorporating paraphrases and label alignment can improve model performance. Performance comparison between Faiss+JS+SR and PLA model indicates leveraging paraphrase raw selection step alone can detect appropriate paraphrases but Siamese network still provides extra bonus.



(a) *Raw Selection Number N* (b) *Pairwise Matching Threshold ρ*

(c) *Shortlisting Skill Length E* (d) *Paraphrase Number K*

Figure 3: *Parameter tuning results on normalized NDCG.*

3.4. Parameter Tuning

Raw selection number N, pairwise matching threshold ρ, Shortlisting skill length E and paraphrase number K are the four major parameters to be tuned in PLA model. For confidential purpose, we report normalized NDCG score using the original NDCG score divided by the largest NDCG score. Observing normalized NDCG changes in Figure 3, the default parameter settings are determined as $N = 100, \rho = 0.8, E = 40, K = 5$.

In Figure 3(a), the model performance always improves when involving more raw paraphrases. With sufficient amount of claimed utterances, selecting Top 100 raw paraphrases can still make positive contributions. Figure 3(b) shows that higher threshold on Siamese network can filter out semantically irrelevant paraphrases to benefit model performance. While the improvement is not prominent with threshold above 0.8. Figure 3(c) indicates that involving more Shortlisting skills can benefit skill retrieval performance. But the positive effect becomes insignificant when skill length is over 20. Figure 3(d) shows a similar trend as Figure 3(a) that more paraphrases lead to better model performance.

4. Conclusion

In this paper, we present a fallback skill retrieval system for unclaimed utterances to recover the conversation between users and SLU systems. By appropriately aligning claimed and unclaimed utterances, our model achieves a boosted performance compared with several strong alternatives. In the future, we will extend the investigation on other label alignment approaches such as teacher-student models and prototypical networks.

5. References

[1] R. De Mori, F. Bechet, D. Hakkani-Tur, M. McTear, G. Riccardi, and G. Tur, "Spoken language understanding," *IEEE Signal Processing Magazine*, vol. 25, no. 3, pp. 50–58, 2008.

[2] K. Clark, M.-T. Luong, C. D. Manning, and Q. Le, "Semi-supervised sequence modeling with cross-view training," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 1914–1925.

[3] Z. Jiang, Z. Gao, J. Lan, H. Yang, Y. Lu, and X. Liu, "Task-oriented genetic activation for large-scale complex heterogeneous graph embedding," in *Proceedings of The Web Conference 2020*, 2020, pp. 1581–1591.

[4] S. Mukherjee and A. Awadallah, "Uncertainty-aware self-training for few-shot text classification," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[5] Y. Wang, S. Mukherjee, H. Chu, Y. Tu, M. Wu, J. Gao, and A. H. Awadallah, "Adaptive self-training for few-shot neural sequence labeling," *arXiv preprint arXiv:2010.03680*, 2020.

[6] T. Sakai, M. C. Plessis, G. Niu, and M. Sugiyama, "Semi-supervised classification based on classification from positive and unlabeled data," in *International conference on machine learning*. PMLR, 2017, pp. 2998–3006.

[7] E. Sansone, F. De Natale, and Z. Zhou, "Efficient training for positive unlabeled learning." *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 11, p. 2584, 2019.

[8] J. Bekker and J. Davis, "Learning from positive and unlabeled data: a survey." *Mach. Learn.*, vol. 109, no. 4, pp. 719–760, 2020.

[9] G. Karamanolakis, D. Hsu, and L. Gravano, "Cross-lingual text classification with minimal resources by transferring a sparse teacher," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, 2020, pp. 3604–3622.

[10] Y. Ziser and R. Reichart, "Deep pivot-based modeling for cross-language cross-domain transfer with minimal guidance," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 238–249.

[11] F. Yang, K. Li, Z. Zhong, Z. Luo, X. Sun, H. Cheng, X. Guo, F. Huang, R. Ji, and S. Li, "Asymmetric co-teaching for unsupervised cross-domain person re-identification." in *AAAI*, 2020, pp. 12 597–12 604.

[12] Z. Gao, H. Li, Z. Jiang, and X. Liu, "Detecting user community in sparse domain via cross-graph pairwise learning," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 139–148.

[13] K. Börner, O. Scrivner, L. E. Cross, M. Gallant, S. Ma, A. S. Martin, L. Record, H. Yang, and J. M. Dilger, "Mapping the co-evolution of artificial intelligence, robotics, and the internet of things over 20 years (1998-2017)," *PloS one*, vol. 15, no. 12, p. e0242984, 2020.

[14] J.-K. Kim and Y.-B. Kim, "Pseudo labeling and negative feedback learning for large-scale multi-label domain classification," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7964–7968.

[15] R. Xu and Y. Yang, "Cross-lingual distillation for text classification," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017, pp. 1415–1425.

[16] Y. Wang, Y. Lin, Z. Gao, and Y. Chen, "A two-stage iterative approach to improve crowdsourcing-based relevance assessment," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3155–3172, 2019.

[17] X. Ma, Y. Shen, G. Fang, C. Chen, C. Jia, and W. Lu, "Adversarial self-supervised data free distillation for text classification," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6182–6192.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[19] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *IEEE Transactions on Big Data*, 2019.

[20] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3973–3983.

[21] Z. Lian, Y. Li, J. Tao, and J. Huang, "Speech emotion recognition via contrastive loss under siamese networks," in *Proceedings of the Joint Workshop of the 4th Workshop on Affective Social Multimedia Computing and First Multi-Modal Affective Computing of Large-Scale Multimedia Data*, 2018, pp. 21–26.

[22] T. Roelleke and J. Wang, "Tf-idf uncovered: a study of theories and probabilities," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008, pp. 435–442.

[23] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. [Online]. Available: https://www.aclweb.org/anthology/P16-1162

[24] W. Ruan, Y. Nechaev, L. Chen, C. Su, and I. Kiss, "Towards an asr error robust spoken language understanding system," *Proc. Interspeech 2020*, pp. 901–905, 2020.

[25] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "Tinybert: Distilling bert for natural language understanding," *arXiv preprint arXiv:1909.10351*, 2019.

[26] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *arXiv preprint arXiv:1509.01626*, 2015.

[27] 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.

[28] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.