Phonetic Embedding for ASR Robustness in Entity Resolution

Xiaozhou Zhou[†], *Ruying Bao*[*], *W. M. Campbell*[†]

[†]Amazon Alexa, United States

{xiaozz, cmpw}@amazon.com

Abstract

Entity Resolution (ER) in spoken dialog systems can suffer from phonetic variation in search queries caused by Automatic Speech Recognition (ASR) errors. In this paper, we propose a phonetic embedding technique to improve the robustness of the ER system to this variation, which includes a phonetic embedding model, a training-data augmentation and sampling method, and an ASR robustness evaluation methodology. We test the technique on two use cases: voice search for videos and for books in the e-commerce domain. Combined with a semantic embedding neural vector search (NVS) model, phonetic embedding reduces the error rate of retrieval by 7.07% relative for video, by 4.23% for books compared to NVS not using phonetic embedding, and by 49.9% for video, and by 35.3% for books compared to a lexical search baseline.

Index Terms: entity resolution, phonetic embedding, ASR robustness

1. Introduction

Entity Resolution (ER) is a downstream component of Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) in a spoken dialog system pipeline. After the utterance is routed to a certain domain, the domain's entity retrieval system takes the corresponding slot value as an input query and searches the index of possible entities to find a match. The ER relevance (search accuracy) depends not only on the search engine itself but also is affected by the quality of the input query. For a voice search system, the queries can be corrupted by upstream errors. One of the major source of errors is ASR which can output wrong, missing, or additional words for the slot value. Besides ASR, query mention variation and segmentation errors from Named Entity Recognition (NER) can also produce similar error patterns. A typical property of these errors is that the incorrect query has similar pronunciation as the ground truth query. Without distinguishing the exact error source, we call this kind of variation in queries "phonetic variation."

There are different techniques we can use to recover errors cased by phonetic variation. In lexical search, character level fuzzy match can find some phonetically similar phrases like "shipping potato" and "chip and potato"; however, it is not able to capture words that sound alike but are written differently (heterographs), like "by/buy/bye", "know/no" and "eye/I." Another method is phonetic search (with n-grams), which uses pronunciation (represented by phonemes) instead of word spelling for matching. It can expand the search space for heterographs or words with similar pronunciation. However, phonetic search has limitation in representing pronunciation, since it cannot tell which phonemes sound more similar than others. For example, "ban", "van", "can" can be converted into phonemes (X-SAMPA [1]) "b { n", "v { n", "k { n" respectively, and they have exactly the same edit distance between each other. However, we know that "ban" and "van" are more likely to be confused in pronunciation (they are both front voiced consonants) than with"can". Phonetic search can also hurt precision by adding more candidates for downstream reranking.

Phonetic embedding, on the other hand, converts a variable-length query into a fixed-dimensional vector based on pronunciation. The similarity of pronunciation is directly reflected by the vector distance. As a neural network method, phonetic embedding can give longer queries a better representation compared to an n-gram search method. The phonetic vectors can also be composed with semantic vectors to be built into the same Neural Vector Search system, thus avoiding exploding the dimension of the candidate space for reranking.

The contribution of this paper includes: (a) we propose an enhanced NVS method with phonetic embedding combined with semantic embedding for improving ASR Robustness in Entity Resolution and show that it can improve the NVS model for ER in video and book domains; (b) we propose an augmentation and sampling method for building phonetic embedding training data from ASR N-best clusters; (c) we create a set of rules to split the ER test data into subsets that contain different types of query variation to test the ASR Robustness of the ER system.

2. Related Work

Most of the previous work on phonetic embedding leverages acoustic features as input. Haque edit distance al.[2] explored a multi-task multi-model for spoken sentence embedding using both speech frames and phonemes as input. Li edit distance al.[3] applied phonetic embedding for the task of speech-driven talking-avatar synthesis. Chen edit distance al.[4] combine phonetic and semantic embedding for spoken content retrieval from speech. Settle edit distance al.[5][6] studied acoustic word embeddings and its application to query-to-example search. Chung and Glass[7] developed a framework for speech-to-vec embedding.

There were also other techniques used for addressing ASR robustness in dialogue systems. Wang edit distance al.[8] proposed an ER system with ASR N-best hypotheses as additional input to the deep learning model. Fazel-Zarandi edit distance al.[9] trained error simulators to generate realistic ASR errors for training more robust NLU models. Raghuvanshi edit distance al.[10] leveraged phonetic and ASR n-best features in the search phase of ER.

The difference of our work with the prior efforts is that our focus is on calculating similarities that are robust to phone variation, and we use phonemes converted by grapheme-to-phoneme (G2P) tool from both the query text and entity catalog as the input of the embedding model. This reduces the dependency on the intermediate output of the ASR system, e. g., phone posteriors, phoneme embeddings, or a lattice that may

[*]This work was done when Ruying Bao worked at Amazon Alexa, US.

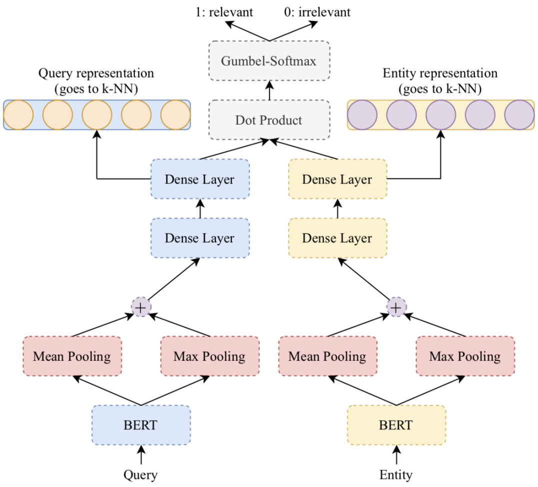

Figure 1: *Siamese Network*
This figure is a sketch of the Siamese Network of both embedding models. For lexical NVS, we use BERT as the encoder(as shown in the figure). For phonetic embedding, the encoder is a bi-LSTM instead.

not be available (esp., for end-to-end ASR systems). A few previous efforts have applied phonetic embedding without acoustic input to NLP tasks, such as Machine Translation[11] and improving ASR robustness in NLU[12]. To our best knowledge, this paper is the first work using phoneme embedding for ER.

3. Model Design

3.1. Neural Vector Search (NVS)

We use a Siamese network [13] to train an ER model and use it as the baseline for pure lexical neural vector search (NVS) model. A Siamese network is a binary classifier used to capture relevance of query and response where their entities are both encoded by a shared model, which is BERT[14] here. We trained semantic embedding (dimension: 768) by fine-tuning the sentence-BERT model pre-trained on both NLI and STSB dataset (first fine-tuned on AllNLI[15], then on STS benchmark training set[16]), then with the domain specific query-response pairs (QRPs).

3.2. Phonetic Embedding Models

Inspired by the success of NVS, we follow the same logic and use phonemes of QRPs. To learn phonetic embedding (dimension: 300), we use a Siamese Network, shown in Figure 1, with 2-layer bi-LSTM models for the sharing encoding model, and a space tokenizer on phoneme sequences. In bi-LSTM, both hidden layers' dimension are 256, followed by a pooling layer and two DNN layers with 300 dimension. The total vocabulary size for phonemes is 50, which is the standard tokenizer across different domains. Other architectures are explored, such as DNN, CNN, and LSTM, but didn't outperform a bi-LSTM (bi-LSTM hyperparameters were tuned on a held-out set).

3.3. Combining Phonetic and Semantic Embedding

To combine phonetic and semantic embeddings, we propose a Weighted Sum model based on [17], which adds phonetic and semantic embeddings with a pre-defined ratio, β. Training semantic and phonetic embeddings separately before feeding into the Weighted Sum model, we fine-tune the semantic embedding from a pretrained BERT model, and train the phonetic embedding from the bi-LSTM from scratch. To map both embeddings to the same dimension, we apply a one-layer DNN to the semantic embedding. For better performance, we also apply a 2-layer DNN to the joint embedding before classification. Figure 2 shows an example for combining semantic and phonetic embedding of the query, “Harry Potter.”

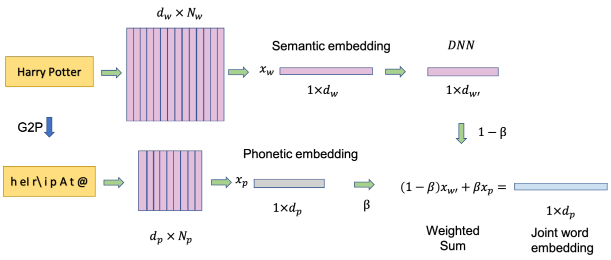

Figure 2: *Architectures of Weighted Sum model*

4. Dataset

4.1. Video Data

4.1.1. Training and Testing Data

We construct the data set for video voice search using search impression feedback. All samples are query-response pairs (QRPs) with their relevancy as labels. If a customer clicked and watched a video on the returned list on screen after the voice query, this QRP will be labeled as a positive pair; otherwise, it's a negative pair. To improve reliability of data sets, we filter positive pairs by calculating click through rate (CTR) defined as number of clicks divided by number of impressions. “Impressions” will be counted once customers see an entity on the search results screen. To reduce noise, we set lower bounds on the CTR value for sampling both the training and test data, and the test data has more strict(higher) CTR threshold than the training data. Additionally, for both sets, we filter out entries for pairs that appeared less than 10 times over the observed period for reducing label noise. For negative pairs, no filtering is used. In the test set, ambiguous queries having multiple ground-truth entities are removed as well.

We randomly sampled about 10 million pairs from search impression data as our training set for semantic and phonetic embedding, including a balanced number of positive and negative pairs. For fine-tuning the joint Weighted Sum model, we randomly sub-sampled one tenth of the training pairs. The test set has several million samples.

4.1.2. Data Augmentation for Training Phonetic Embedding

Besides the QRPs from the search impression data, we also constructed positive and negative pairs from ASR N-best for phonetic embedding training. For our ASR systems, we have the top recognition result and also a list of N-best results that can be further processed by downstream systems. We used up to 5 N-best output from ASR in our experiments.

Hard Sampling. Considering our main purpose of using phonetic embedding in ER is to make it robust to ASR errors, we construct the positive pairs with phrases that tend to be confused by ASR, which can be found in the ASR N-best output clusters. Each cluster contains at most 5 output text with the highest confidence scores from ASR for a certain query. For example, we can select any two different phrases from the ASR N-best cluster,

"bailey two", "bailey to", "bayley two", "bailee two"

as the positive pair. An upper limit threshold for phoneme edit distance normalized by length is used to make sure that the positive pairs are similar enough in pronunciation. The negative pairs are sampled by randomly selecting phrase pairs from different ASR N-best clusters which have no overlap, and set both lower and upper threshold for the normalized phoneme edit distance. In practice, we use MinHash [18] to cluster the data first and sample negative pairs from the clusters in order to speed up the process. To avoid hard sampling that is too difficult and may confuse the model, we guarantee that there is a gap between positive pairs' upper bound and negatives' lower bound, which also ensures negative pairs of a certain hardness level.

Pruning Method. In hard sampling from ASR N-best, we can't guarantee that mentions in positive and negative pairs correspond to the same entity, and the assumption that mentions in each ASR N-best cluster always have the same ground truth entity is not correct. So we use catalog to further filter mentions in ASR N-best clusters to learn more about catalog entity, so that ER-no-match rate may be decreased. Here is the logic to generate positive and negative pairs:

- Positive pairs: sample from the same ASR N-best clusters and guarantee that at least one mention has the ground-truth entity in catalog.
- Negative pairs: belong to different ASR N-best clusters.

In our sampling process, we set positive pairs with maximum of normalized edit distance as 0.3 and negative pairs with minimum of normalized edit distance as 0.5. We notice that phonetic embedding trained with 100% ASR N-best data has more than 20% performance drop compared to that trained with QRPs. For including both phonetic and semantic information, we mix QRPs and ASR data with various ratio and 10% ASR data mixed with 90% QRPs works the best.

4.2. Book Data

For book voice search, we constructed the data set using query reformulation signals. For example, if the request was first recognized by ASR as "Harry Pot" and the ER system does not return the desired book, they may repeat it within a small time window, and this time it is recognized as "Harry Potter" which produces the desired Harry Potter book, we can construct the query in the first turn "Harry Pot" and the returned entity for the second turn as a positive QRP. The negative QRPs will be sampled randomly from the catalog just as in Video domain. We didn't do data augmentation for book domain.

The test data for book is sampled from the reformulation data with er-no-match for the first turn, i.e., the ER system returned no entities. These are the queries for which the lexical search system works poorly. Using these queries as the test data we can directly see how much improvement semantic or phonetic embedding can achieve from the lexical search baseline.

5. Experiments

5.1. Evaluation Methods

In order to test the ASR robustness of ER system under different query variation types, we used rules to classify test cases by comparing the query value and the labeled entity value. The resulted subsets are mutually exclusive and complete, and the priority of classification are ordered as the following:

- **No variation**: query and entity are exact lexical match (case insensitive)
- **Phonetic variation**: phoneme edit distance between query and entity values $<= 5$, which is further classified as:
  - **Spoken-written variation**: entity value can be converted to query using written-to-spoken tokenizer
  - **Heterographs**: same phonemes (generated by G2P tool), different text
  - **Phone variation**: $1 <=$ phoneme edit distance between query and entity $<= 5$
- **Lexical variation**: phoneme edit distance between query and entity values > 5, which is further classified as:
  - **Over-specification**: query has more words than the entity
  - **Under-specification**: query has fewer words than the entity
  - **Word variation**: other cases with phoneme edit distance between query and the entity > 5
- **No match**: no ground truth entity value

ASR Robustness methods are expected to have most improvement on the phonetic variation subsets.

5.2. Test Results

For both the semantic and phonetic embedding models, we apply the embedding vectors to the ER retrieval task, and use FAISS [19] to run the vector search. Video and Book search results are in Table 1 and Table 2 respectively. We compare these different techniques:

- **baseline(lexical search)**: retrieve top fifty candidates by lexical search and rerank by importance scores showing the videos or books' popularity.
- **phonetic search**: retrieve top fifty candidates by lexical plus phonetic search and rerank by importance scores.

  Both lexical search and phonetic search are conducted using standard information retrieval methods with Elasticsearch.
- **NVS**: retrieve top fifty candidates by pure semantic embedding (NVS) and lexical search separately, and rerank by importance scores.
- **NVS+phonetic**: retrieve top fifty candidates by the joint semantic+phonetic embedding trained with QPRs only, and lexical search separately, and rerank by importance scores.

- **NVS+phonetic+data augmentation**: retrieve top fifty candidates by the joint semantic+phonetic embedding trained with mixing of 10% ASR N-best augmented data and 90% QRPs, and lexical search separately, and rerank by importance scores. The thresholds of normalized edit distance used in augmentation data sampling are chosen as: positive upper bound is 0.3 and negative lower bound is 0.5. This technique was tested for Video data only.

All results are in relative error reduction (with baseline to be 0) measured by $recall@5$. We also tested the metrics on the phonetic variation subsets to show specifically how these techniques perform on queries that are most likely to have ASR variation. The number and percentage of samples in each subset is recorded in the top rows. Note that ”Phonetic variation” is the sum of the following 3 columns: ”Heterographs”, ”Spoken-written variation”, and ”Phone variation”.

Table 1: *Relative error reduction for Video voice search*

Technique	**Whole data set**	**Phonetic variation**	Spoken-written variation	Heterographs	Phone variation
	6,934,201 (100%)	1,102,661 (15.90%)	138,601 (2.00%)	232,900 (3.36%)	731,160 (10.54%)
baseline(lexical search)	0	0	0	0	0
phonetic search	34.7%	52.3%	**89.9%**	55.3%	41.6%
NVS	39.7%	60.3%	74.7%	50.8%	58.7%
NVS+phonetic	44.0%	71.2%	85.7%	65.0%	68.8%
NVS+phonetic +data augmentation	**49.9%**	**79.0%**	88.6%	**89.3%**	**73.9%**

Table 2: *Relative error reduction for Book voice search*

Technique	**Whole data set**	**Phonetic variation**	Spoken-written variation	Heterographs	Phone variation
	11,224 (100%)	966 (38.11%)	0 (0%)	10 (0.39%)	956 (37.71%)
baseline(lexical search)	0	0	0	0	0
phonetic search	34.1%	44.7%	N/A	70.0%	44.5%
NVS	32.5%	48.7%	N/A	70.0%	44.1%
NVS+phonetic	**35.3%**	**55.5%**	N/A	**80.0%**	**55.5%**

5.3. Discussion

The experimental results show that phonetic information does help the retrieval tasks, especially on phonetic variation queries. Also the Weighted Sum model is effective for combining both semantic and phonetic information which performs better than both the pure semantic embedding model and phonetic search. With data sets generated containing ASR N-best information, we can further improve phonetic embedding performance and preserve that information when it is combined with the semantic embedding. We notice that both Hard Sampling and the Pruning Method with (normalized) edit distance outperform the standard QRPs, and normalized edit distance works better than edit distance in most subsets. Among those combinations, the pruning method with normalized edit distance does the best job on both ASR variation and lexical variation, whose result is shown in the table.

Here are two interesting findings: 1. the Pruning method with normalized edit distance outperforms other methods in heterographs, at least by 10%; 2. Hard Sampling with normalized edit distance outperforms other methods in phone variation. Diving deep into heterographs, we split the test set into two types: segment variation and character variation. Segment variation includes samples their ground-truth entity title has the same character string as the query but different space position, such as ”Rocketman” and ”Rocket man”; and character variation includes those having different character strings. We notice that the Pruning method with normalized edit distance effectively improves performance on segment variation, which may due to the higher frequency of catalog entity titles during phonetic model training, and the accuracy on character variation maintains the same. Combining with observations in phone variation, we believe normalized edit distance is a better metric to evaluate phonetic similarity.

6. Conclusions and Future Work

In this paper, we developed a phonetic embedding model based on Siamese Network and trained with phoneme pairs sampled from both QRPs and ASR N-best data. We combined the phonetic embedding with a semantic embedding trained with lexical features, and applied this model to Video and Book ER. To measure the effectiveness of the method, we proposed a query variation classification method to create subsets of the test data with different variation types.

In the evaluation of retrieval tests, we see that compared to lexical search baseline, phonetic embedding reduces the error rate in phonetic variation subset by 71.2% and all test set by 44.0% in Video domain, and reduces the error rate in phonetic variation subset by 55.5% and all test set by 35.3% in the book domain; With ASR N-best data augmentation, we further reduces the error rate in phonetic variation by 79.0% and all test set by 49.9% in the Video domain. Also, the Pruning method improves phonetic embedding performance on retrieval tasks by adding more observation on catalog entity titles during training.

As future work, we plan to do the following: (a) Explore better data sampling and model structures for phonetic embedding, for instance, using BERT[14] or more advanced transformer based models[20][21] for phonetic embedding; (b) Instead of training the semantic and phonetic embedding models separately, we will study the dual-encoder model for joint training of both embeddings at the same time; (c) Study the fusion of different retrieval results from phonetic and semantic embedding, lexical search, and other retrieval methods such as phonetic search, which may involve calibration of the retrieval scores and post-ER ranking.

7. Acknowledgement

The authors would like to thank Chris Qiu for helping us processing the data sets, building the model training pipeline and reviewing the experiment results, Yaroslav Nechaev for providing the NVS baseline model, and also Haoyu Wang for helpful discussion.

8. References

[1] J. C. Wells, “Computer-coding the IPA: a proposed extension of SAMPA,” 1995.

[2] A. Haque, M. Guo, P. Verma, and L. Fei-Fei, “Audio-linguistic embeddings for spoken sentences,” 2019.

[3] X. Li, Z. Wu, H. Meng, J. Jia, X. Lou, and L. Cai, “Phoneme embedding and its application to speech driven talking avatar synthesis,” in *INTERSPEECH*, 2016.

[4] Y.-C. Chen, S.-F. Huang, C.-H. Shen, H. yi Lee, and L. shan Lee, “Phonetic-and-semantic embedding of spoken words with applications in spoken content retrieval,” 2019.

[5] S. Settle, K. Levin, H. Kamper, and K. Livescu, “Query-by-example search with discriminative neural acoustic word embeddings,” in *INTERSPEECH*, 2017.

[6] S. Settle and K. Livescu, “Discriminative acoustic word embeddings: Recurrent neural network-based approaches,” *2016*

IEEE Spoken Language Technology Workshop (SLT), pp. 503–510, 2016.

[7] Y.-A. Chung and J. R. Glass, “Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech,” in *INTERSPEECH*, 2018.

[8] H. Wang, J. Chen, M. Laali, J. King, K. Durda, W. M. Campbell, and Y. Liu, “Leveraging asr n-best in deep entity retrieval,” in *Interspeech 2021*, 2021. [Online]. Available: https://www.amazon.science/publications/leveraging-asr-n-best-in-deep-entity-retrieval

[9] M. Fazel-Zarandi, L. Wang, A. Tiwari, and S. Matsoukas, “Investigation of error simulation techniques for learning dialog policies for conversational error recovery,” 2019. [Online]. Available: https://arxiv.org/abs/1911.03378

[10] A. Raghuvanshi, V. Ramakrishnan, V. Embar, L. Carroll, and K. Raghunathan, “Entity resolution for noisy ASR transcripts,” 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): System Demonstrations*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 61–66. [Online]. Available: https://aclanthology.org/D19-3011

[11] H. Liu, M. Ma, L. H. 0001, H. Xiong, and Z. He, “Robust neural machine translation with joint textual and phonetic embedding,” in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, A. Korhonen, D. R. Traum, and L. Màrquez, Eds. Association for Computational Linguistics, 2019, pp. 3044–3049. [Online]. Available: https://www.aclweb.org/anthology/P19-1291/

[12] A. Fang, S. Filice, N. Limsopatham, and O. Rokhlenko, *Using Phoneme Representations to Build Predictive Models Robust to ASR Errors*. New York, NY, USA: Association for Computing Machinery, 2020, p. 699–708. [Online]. Available: https://doi.org/10.1145/3397271.3401050

[13] S. Chopra, R. Hadsell, and Y. LeCun, “Learning a similarity metric discriminatively, with application to face verification,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1, 2005, pp. 539–546 vol. 1.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *NAACL*, 2019.

[15] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, “A large annotated corpus for learning natural language inference,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[16] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, “Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation,” *arXiv preprint arXiv:1708.00055*, 2017.

[17] H. Liu, M. Ma, L. Huang, H. Xiong, and Z. He, “Robust neural machine translation with joint textual and phonetic embedding,” *arXiv preprint arXiv:1810.06729*, 2018.

[18] B. Rao and E. Zhu, “Searching web data using minhash lsh,” in *Proceedings of the 2016 International Conference on Management of Data*, 2016, pp. 2257–2258.

[19] J. Johnson, M. Douze, and H. Jegou, “Billion-scale similarity search with GPUs,” *arXiv*, 2017. [Online]. Available: https://arxiv.org/pdf/1702.08734.pdf

[20] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “ALBERT: A lite BERT for self-supervised learning of language representations,” *ArXiv*, vol. abs/1909.11942, 2020.

[21] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” *ArXiv*, vol. abs/1907.11692, 2019.