Leveraging ASR N-best in Deep Entity Retrieval

Haoyu Wang[1], *John Chen*[2†], *Majid Laali*[3], *Kevin Durda*[3], *Jeff King*[3], *William Campbell*[1], *Yang Liu*[1]

[1]Amazon Alexa, United States
[2]University of Toronto
[3]Amazon Alexa, Canada

{wanhaoyu, laalim, durdak, jfkin, cmpw, yangliud}@amazon.com
johnc@cs.toronto.edu

Abstract

Entity Retrieval (ER) in spoken dialog systems is a task that retrieves entities in a catalog for the entity mentions in user utterances. ER systems are susceptible to upstream errors, with Automatic Speech Recognition (ASR) errors being particularly troublesome. In this work, we propose a robust deep learning based ER system by leveraging ASR N-best hypotheses. Specifically, we evaluate different neural architectures to infuse ASR N-best through an attention mechanism. On 750 hours of audio data taken from live traffic, our best model achieves 11.07% relative error reduction while maintaining the same performance on rejecting out-of-domain ER requests.

Index Terms: speech recognition, error correction, entity retrieval, N-best, self-attention

1. Introduction

In a spoken dialogue system, ASR takes the audio of a user request and generates hypotheses in the form of text. The Natural Language Understanding (NLU) system, consisting of domain classifiers (DC), intent classifiers (IC), and named entity recognizers (NER), interprets and extracts useful information based on the ASR hypotheses. Entity mentions or slots generated by the named entity recognizers will be passed as input queries to the downstream entity retrieval (ER) system. The ER component searches the catalog based on the intent and entity type to retrieve the most relevant real world object for the entity mention. Consider a speech-based virtual assistant that helps users control smart home appliances. For an utterance *turn on the den room light*, NER labels *den room light* as *Appliance*, and then ER takes this as a query and links to an entity in the catalog inventory, returning the most relevant smart light. Since "den" is a rare word, ASR may mistranscribe "den" as "dining", and the ER system in turn may end up retrieving nothing or an irrelevant entity from the catalog, adversely affecting the customer experience. Therefore, making ER robust to ASR errors is important for the overall spoken dialog system performance.

Despite recent advancements in ASR systems [1], accurately transcribing infrequent domain-specific words remains a challenge since many of these words are rare or out-of-vocabulary for a general-purpose ASR system. Although we may enlarge the vocabulary of the ASR system, it may still fail to rank those infrequent terms as the top hypothesis during the decoding phase. Such shortcoming has inspired a substantial amount of efforts to make each stage of a spoken dialogue system robust to ASR errors [2, 3, 4, 5].

Although ASR usually generates a list of ranked hypotheses during beam search decoding, i.e., N-best hypotheses, most downstream components only leverage the top-ranked ASR hypothesis as input and ignore the additional hypotheses that may contain valuable information for correcting ASR errors. For the previous example, we may find the correct term "den" in ASR's N-best hypotheses, although not ranked at the top. In this work, we thus propose to leverage additional information from the ASR N-best hypotheses for ER in spoken dialog systems. We compare several different model architectures and propose a generalized attention-based mechanism to infuse additional ASR N-best hypotheses during the query encoding phase for deep ER systems. Using data of 750 hours audio from live traffic, we demonstrate the effectiveness of leveraging ASR N-best information in the ER task – our proposed method yields 11.07% error reduction for the in-domain ER request, without degrading performance for out-of-domain ER requests.

2. Related Work

Various methods have been explored to alleviate the impact of ASR errors in spoken dialogue systems. Some focused on recovering those errors within ASR. For instance, [6] proposed an enhanced domain-specific language model to increase the vocabulary diversity for domain-specific words. Post-editing models have also been explored to rewrite the error-prone ASR transcriptions [3, 7]. Various ranking models have been proposed to leverage word and contextual information to rerank hypotheses from ASR output [8]. In dialogue systems, error simulators have been trained to generate realistic errors including ASR errors for NLU model training in order to build models robust to noisy ASR output [2]. There is also some previous work that is more relevant to the ER task that we tackle. [4] tried to recover ASR errors for ER by introducing phonetic features during the search phase. In [5], a transformer based model was proposed to explicitly rewrite the noisy search term based on synthetic data generated by error simulation. While these techniques are effective in mitigating the detrimental effect of ASR errors by reducing the overall word error rate, they do not leverage information available in the ASR N-best.

It has often been shown that there can be useful information at lower ranked positions in the ASR N-best list for a given utterance. One approach that is frequently applied is to rely on meta-features to rerank the hypotheses [9, 10, 11]. [12] constructed a word confusion network on the N-best hypotheses and used it to conduct NER, but it is constrained with rules on limited use cases. In [13, 14], the N-best list was modeled jointly through a probabilistic approach without re-ordering. In [15], a Bi-LSTM model was proposed to leverage ASR N-best on a domain classification task. While neural network based ER has become the state-of-the-art approach, to the best of our knowledge there hasn't been any study conducted to leverage

†Work was done during internship at Amazon Alexa, Canada

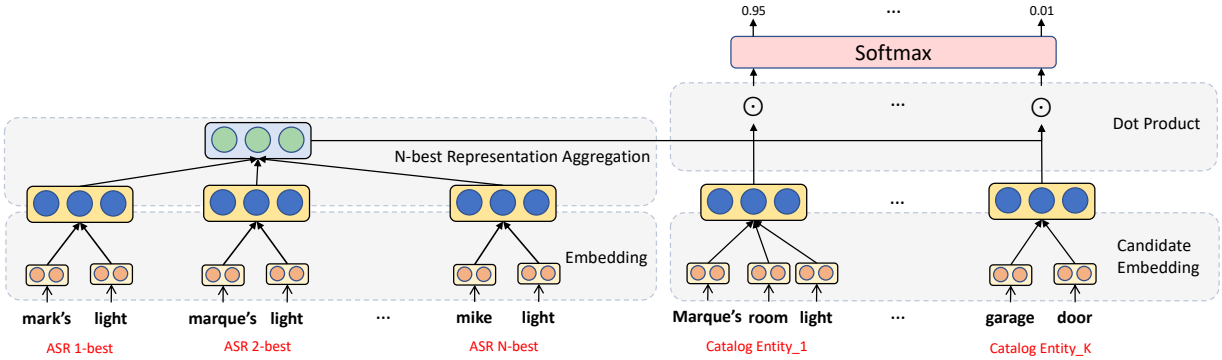

Figure 1: *Dual encoder architecture for ASR N-best ER model. All ASR N-best mentions will be aggregated into a single representation. Each candidate entity will be encoded individually and the score will be calculated by a dot product with ASR N-best representation.*

ASR N-best in neural network models for ER.

3. Proposed Method

To formalize the task of ER, we define a query as $m = \{t_1, t_2, ..., t_N\}$ where t_i is the i-th token after applying tokenization on the entity mention in the user utterance. The task of ER is to retrieve a proper entity from a catalog defined as $C = \{c_1, c_2, ..., c_K\}$, or reject the query as out of domain input. A given catalog entity c_i can be presented as $c_i = \{t_1, t_2, ..., t_M\}$. To solve the problem of ER, we design a model to learn the score (similarity) between a given pair as $s(m, c_i) \in [0, 1]$, and then pick the top-1 ranked c_i as the result or reject the query if the top score is below a threshold θ.

As we aim to leverage ASR N-best information to improve the robustness of ER against ASR errors, we will expect a list of mentions available as $M = \{m_1, m_2, ..., m_L\}$ from ASR N-best. These can be easily obtained by aligning ASR hypotheses and mapping the entity mention in the top hypothesis to others, or performing NER on each hypothesis. The goal of our model is thus to learn the score (similarity) between a list of mentions and an entity candidate c_i, $s(\{m_1, m_2, ..., m_L\}, c_i) \in [0, 1]$.

Although different neural network architectures have been proposed for entity retrieval and linking [16, 17, 18], in this work we follow the commonly used dual encoder neural network architecture in [18]. We believe our proposed approach to encode ASR N-best information could also easily generalize to other neural network architectures. For the dual encoder model, we first encode the query by an encoder $\mathbf{h}^M = f(\{m_1, m_2, ..., m_L\}) \in R^d$ and then encode the catalog entity by another encoder $\mathbf{h}^{c_i} = g(c_i) \in R^d$. We then define the score similarity function between $\mathbf{h}^M$ and $\mathbf{h}^{c_i}$ as their dot product. While we could have used other mechanisms that jointly model the two representations [16, 17], we mainly focus on dual encoder models because they will allow us to scale to search over millions of entities using an efficient k-nearest neighbour search [19, 20]. Figure 1 presents an overview of the model architecture. The rest of this section will focus on how we generate $\mathbf{h}^M$ and $\mathbf{h}^{c_i}$ from queries and entities.

3.1. Text Encoding

While there have been multiple different approaches proposed for text encoding, we choose to follow the simple yet powerful approach of embedding pooling [21, p. 106]. While our approach can also be applied on other advanced text encoders, we do not focus on that in our study. For a given token t_i, we first conduct an embedding look up as $\mathbf{h}^{t_i} \in R^d$. Hence for a given sequence of tokens $t = \{t_1, t_2, ..., t_N\}$, we can obtain $emb(\{t_1, t_2, ..., t_N\}) = \{\mathbf{h}^{t_1}, \mathbf{h}^{t_2}, ..., \mathbf{h}^{t_N}\} \in R^{N \times d}$. We further conduct an average over the N embeddings to get the text-level representation as $\mathbf{h}^t = 1/N \times \sum_1^N \mathbf{h}^{t_i}$.

This encoding approach can be applied to both the query and the candidate entity, using either shared embedding or separately learned embedding, resulting in $\mathbf{h}^{m_i}$ and $\mathbf{h}^{c_j}$ respectively.

3.2. Mention Representation Aggregation

We evaluate different methods to aggregate $\mathbf{h}^{m_i}$ from ASR N-best output to compute $\mathbf{h}^M$ ($M = \{m_1, m_2, ..., m_L\}$).

• **Pooling**: One straight forward way of computing $\mathbf{h}^M$ is to conduct an average pooling of $\mathbf{h}^{m_i}$ as $\mathbf{h}^M = 1/L \times \sum_1^L \mathbf{h}^{m_i}$.

• **Learned Weighted Combination**: The Pooling approach above gives the same weight to different ASR hypotheses. Since hypotheses have different likelihoods (or confidence score), we introduce L learnable parameters defined as $\alpha_i \in [0, 1]$ ($i \in \{1, 2, ..., L\}$) to weigh each representation or hypothesis. Then we compute the query representation as $\mathbf{h}^M = \sum_1^L \alpha_i \times \mathbf{h}^{m_i}$.

• **Global Attention**: Rather than using a set of global weights that are only dependent on the rank of a particular hypothesis on the n-best list, we propose to learn weights dependent on the representation of each ASR N-best hypothesis. In light of work [22], we introduce a learnable context vector $\mathbf{c} \in R^d$ that is used to attend to each mention's representation to calculate the weighted score as $\alpha_i = \mathbf{c} \cdot \mathbf{h}^{m_i}$. After obtaining these weights, we follow the same weighted averaging approach to compute $\mathbf{h}^M$.

• **Concatenated Projection**: Another way to enable the model to automatically decide the aggregation of multiple representations is to first concatenate all the representations $\mathbf{h}^{m_i}$ as $\mathbf{h}^{concat} = [\mathbf{h}^{m_1}, \mathbf{h}^{m_2}, ..., \mathbf{h}^{m_L}]$, then apply a dense layer to project it back to the model dimension as $\mathbf{h}^M = W\mathbf{h}^{concat}$, where $W \in R^{Ld \times d}$.

• **Self-Attention**: In this work, we propose to apply self-attention mechanism [23] to aggregate all the mention representations. Using the L representations $H = [\mathbf{h}^{m_1}; \mathbf{h}^{m_2}; ...; \mathbf{h}^{m_L}]$, we first compute the self-attention scores $A \in [0, 1]^{L \times L}$ by:

$$A = \text{softmax}\left(\frac{(QH)(KH)^T}{\sqrt{d}}\right) \quad (1)$$

where $Q, K \in \mathbb{R}^{d \times d}$ are the parameters of the model. We then compute a new sequence of representations $\hat{H} = [\hat{\mathbf{h}}^{m_1}; \hat{\mathbf{h}}^{m_2}; ...; \hat{\mathbf{h}}^{m_L}]$ by:

$$\hat{H} = AVH \tag{2}$$

where $V \in \mathbb{R}^{d \times d}$ are the parameters of the model and A is the previously computed attention score matrix.

After this we can again follow the pooling approach to average all the representations $\hat{\mathbf{h}}^{m_i}$ as $\mathbf{h}^M = 1/L \times \sum_1^L \hat{\mathbf{h}}^{m_i}$. Since the newly introduced parameters can help learn a proper relation between given mentions $\mathbf{h}_i$ and $\mathbf{h}_j$ based on the representation themselves to guide the aggregation, we expect this to be more powerful than the aforementioned approaches.

3.3. Score and Threshold

Once we obtain the query representation $\mathbf{h}^M$ and a given candidate entity's representation $\mathbf{h}^{c_i}$, we then compute the dot product between the two as a raw similarity score. A softmax function will then be applied to normalize that into probabilities. For the ER task we would like to optimize the model to assign 1 to the correct entity and 0 to the rest of the entities in the catalog. Since in a spoken dialog system, one may ask something out of the catalog, we should ideally reject the query instead of returning something wrong. To address that, a tuned threshold θ can be used to reject any top-ranked entity which has a score below the threshold, that is, the final system decision is:

$$\hat{y} = \begin{cases} \text{argmax}_i(\mathbf{h}^M \cdot \mathbf{h}^{c_i}) & \text{if softmax}(\mathbf{h}^M \cdot \mathbf{h}^{c_i}) \geq \theta \\ -1 & \text{else} \end{cases} \tag{3}$$

4. Dataset

4.1. Dataset Construction

We conduct our experiment on 750 hours of de-identified utterances that users request to control a smart device. Each utterance contains an appliance name derived by the upstream ASR+NLU components, and ER needs to retrieve the right entity from that user's registered smart devices. The ground-truth of the utterance was generated through user feedback signal and annotation process. For instance, given a user asking "turn on my bedroom light", the "bedroom light" will be identified as an appliance name and sent for ER to get the actual entity from the user's registered devices. If ER returns the correct user-defined entity "My Bedroom Light" and the system receives a positive feedback, we consider "My Bedroom Light" as the ground-truth for the query "bedroom light". If a user asks "turn on bedroom light", and ASR accidentally mis-transcribes that into "bathroom light" due to background noises or other issues, ER will fail to retrieve any entity for "bathroom light", and the system may receive a negative feedback signal. In some cases the user may immediately repeat or reformulate the query, and get the right transcription (i.e., "turn on bedroom light") and correct retrieved entity by ER ("My Bedroom Light"). If both queries happen consecutively in a short interval, we will then associate "Bedroom Light" as the ground-truth for the mis-transcribed "bathroom light" in the first utterance. This approach provides us in-domain utterances. We also annotate out-of-domain utterances when there is no correct entity that can be matched with the given appliance name, and hence the ground-truth should be empty. For example, consider an utterance, "turn off the music", due to NLU errors, "the music" has been tagged as an appliance name and passed to ER. Such utterances are collected to form an out-of-domain data set.

4.2. Dataset Processing

For each utterance, we obtain and process the ASR N-best hypotheses by aligning additional ASR N-best hypotheses to ASR 1-best hypothesis token-by-token using Levenshtein approach [24]. Based on the alignment, we can then infer the alternative mentions in other ASR hypotheses based on the mention in the 1-best hypothesis. Note that this string alignment is a much more computationally efficient method compared to running NER on every hypothesis. After this processing, for each utterance we have a maximum of 5 mentions (this is because of the beam search limit in the ASR decoder). It is worth noting that the mentions may be the same in different ASR hypotheses. We keep these duplicates as is since we believe this is a valuable information for the model to learn how to weigh each ASR hypothesis. We further process the mentions and each catalog entity's name with a sentence-piece tokenizer [25] trained on this dataset with a vocabulary size of 6000 and use that for all experiments.

5. Experiments

5.1. Experiment Setup

In this work, we use the same hyper-parameters for all the models. We set the model dimension $d = 128$. We do not share the embeddings between the mention encoder and candidate encoder. The training is optimized using Adam optimizer with a learning rate of 0.001 and batch size of 128. We set the maximum number of training epochs to be 1000 with an early stopping strategy based on the development dataset's loss not decreasing for 10 epochs. We didn't exhaustively tune the hyper-parameters of the network architecture as it is not the main focus of the study.

Since each user may have registered a different number of smart devices, during training we randomly pick at maximum 10 negative entities from registered devices to form the negative cases for a given utterance for model training. During inference time, we calculate the scores on the entire set of entities in the catalog to find the best matched entity.

5.2. Baseline

In this work, one straight forward baseline is to use the same text encoding to encode only the mention from ASR top-1 hypothesis as $h^M = h_1$. This can serve as a baseline to help us understand the improvement from using ASR N-best. We evaluate our proposed self-attention based approach against this baseline as well as other methods discussed in Section 3.

5.3. Experiment Results on In-domain Dataset

Because of the nature of our spoken dialog system application domain, for a given user query there will be only one relevant entity in the catalog and all the other entities are not relevant. Hence the most critical metric is the correctness of the top entity that the ER system returns. We thus evaluate different approaches using the accuracy of the system, which is essentially the Precision@1 of a retrieval problem[26], defined as:

$$\text{Accuracy} = \frac{\sum_{i=1}^{N_{\text{id}}} \mathbb{1}(\hat{y}_i == y_i)}{N_{\text{id}}} \tag{4}$$

where $\hat{y}_i$ denotes the system's output (Equation (3)) for the i-th example in the dataset, y_i denotes the index of the ground-truth, and N_{id} is the number of examples in the in-domain dataset. It

is worth noting that $\hat{y}$ depends on the threshold θ of the model. We set $\theta = 0$ for this experiment since the dataset is in-domain (i.e., there exists a ground-truth entity for each utterance), and leave the discussion of varying θ in the next section.

Table 1: *Experiment results for the in-domain data. The Relative Error Reduction for a model m is calculated by comparing the relative difference between* $(100\% - Accuracy_m)$ *and* $(100\% - Accuracy_{baseline})$.

	Relative Error reduction(%)
Baseline w/o ASR N-best	0.00
Mean Pooling	6.02
Learnable Weights	8.61
Global Attention	8.57
Concatenation	8.87
Self-Attention	**11.07**

Table 1 shows the experimental results for the in-domain dataset. By comparing all the methods that leverage ASR N-best against the baseline, we notice significant relative error reduction, which proves the importance of using ASR N-best in ER tasks. By further comparing the different proposed approaches, we can see that the mean pooling approach performs the worst, since it has no learnable parameters but merely averages representations from different ASR hypotheses. Introducing additional learnable weights to calculate weighted averages of the representations is better than mean pooling, but due to the limitation of weak linear transformation, it under-performs the best approach. For global attention approach, although more learnable parameters are introduced comparing to the the learnable weights approach, it doesn't show much improvement. For the concatenation approach, although it has the highest number of additional parameters and can theoretically learn such a weighting strategy, it still under-performs the self-attention based approach, potentially due to the lack of explicit modeling of the importance of each mention. The best performing approach is the self-attention model architecture (a relative error reduction of 11.07% over the baseline), which enables the model to dynamically decide the importance of each representation by attending to other representations.

5.4. Experiment Results About Thresholds

While the experiments above demonstrate the improvement from leveraging the ASR N-best, we conducted additional experiments that are more similar to the real scenario where the ER system may receive and should reject an out-of-domain query when a user asks something irrelevant to the dialog system. As described in Section 3.3, we apply a threshold θ to reject a given query if the top-ranked entity's confidence is below that threshold. To evaluate the effectiveness of rejecting out-of-domain examples, on the out-of-domain dataset, we again measure the Accuracy of a model's output:

$$\text{Accuracy} = \frac{\sum_{i=1}^{N_{\text{ood}}} \mathbb{1}(\hat{y}_i == y_i)}{N_{\text{ood}}} \quad (5)$$

where N_{ood} is the number of examples in the out-of-domain dataset.

With the introduction of θ, the system may incorrectly reject an in-domain utterance, which will have a negative impact on the accuracy. To understand if the performance improvement as observed for the in-domain data can still hold, we present results for both the in-domain and the out-of-domain dataset using different values of θ, as shown in Figure 2. By looking at the Relative Error Reduction of our best-performing self-attention approach against the baseline, we notice that our proposed approach outperforms the baseline consistently with a large margin on accuracy for in-domain dataset for any given threshold. For out-of-domain dataset, we observe nearly the same accuracy curves for any threshold θ, which demonstrates that our approach can still maintain the same level of performance in rejecting out-of-domain examples. Although we notice better performance for rejecting out-of-domain cases using a larger threshold (e.g., $\theta = 0.9$), practically we probably will not choose such a threshold because of the significant drop of the accuracy for the in-domain dataset. A proper threshold can be chosen depending on the distribution and the trade-off for the two cases.

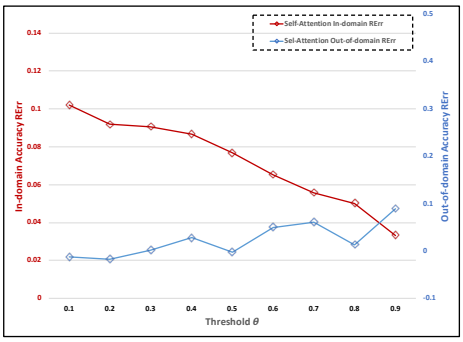

Figure 2: *Relative Error Reduction when varying threshold* θ.

5.5. Discussions

Through the experiments, we compare several state-of-the-art approaches used for aggregating the ASR N-best representations and observe that our proposed self-attention based approach performs the best for the ER task. While it is intuitive to think that the ASR N-best may contain important information to make the representation more robust to ASR errors, it is also a key to appropriately encode that information for the model to obtain the maximum benefit. To make the most use of ASR N-best, the model should learn that it can rely on the ASR 1-best hypothesis for a substantial amount of cases but also should learn from alternative hypotheses in the ASR N-best list via proper weighting.

6. Conclusion

In this paper, we demonstrate the effectiveness of leveraging ASR N-best information in neural network models for the ER task. We compare different model architectures and propose a generalized attention-based mechanism to infuse additional ASR N-best hypotheses via proper representation weighting during the query encoding phase for deep ER system, which yields a relative error reduction of 11.07% while maintaining the same performance in rejecting out-of-domain ER requests. Our approach to aggregate the representations from ASR N-best can be applied to other retrieval and ranking models for ER or general encoders for other NLU downstream tasks.

7. References

[1] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[2] M. Fazel-Zarandi, L. Wang, A. Tiwari, and S. Matsoukas, "Investigation of error simulation techniques for learning dialog policies for conversational error recovery," *ArXiv*, vol. abs/1911.03378, 2019.

[3] M. Li, W. Ruan, X. Liu, L. Soldaini, W. Hamza, and C. Su, "Improving spoken language understanding by exploiting asr n-best hypotheses," *arXiv preprint arXiv:2001.05284*, 2020.

[4] 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*, 2019, pp. 61–66.

[5] H. Wang, S. Dong, Y. Liu, J. Logan, A. Agrawal, and Y. Liu, "Asr error correction with augmented transformer for entity retrieval," in *INTERSPEECH*, 2020.

[6] J. Guo, T. N. Sainath, and R. J. Weiss, "A spelling correction model for end-to-end speech recognition," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5651–5655.

[7] Z. Xie, A. Avati, N. Arivazhagan, D. Jurafsky, and A. Y. Ng, "Neural language correction with character-based attention," 2016.

[8] H. Sak, M. Saraçlar, and T. Güngör, "Discriminative reranking of asr hypotheses with morpholexical and n-best-list features," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, 2011, pp. 202–207.

[9] A. Chotimongkol and A. I. Rudnicky, "N-best speech hypotheses reordering using linear regression," in *INTERSPEECH*, 2001.

[10] M. Rayner, D. Carter, V. Digalakis, and P. Price, "Combining knowledge sources to reorder n-best speech hypothesis lists," *ArXiv*, vol. abs/cmp-lg/9407010, 1994.

[11] M. Gabsdil and O. Lemon, "Combining acoustic and pragmatic features to predict recognition performance in spoken dialogue systems," in *ACL*, 2004.

[12] D. Hakkani-Tür, F. Béchet, G. Riccardi, and G. Tur, "Beyond asr 1-best: Using word confusion networks in spoken language understanding," *Computer Speech & Language*, vol. 20, no. 4, pp. 495–514, 2006.

[13] J. Williams, "Exploiting the asr n-best by tracking multiple dialog state hypotheses," in *INTERSPEECH*, 2008.

[14] C. Lee, S. Jung, and G. Lee, "Robust dialog management with n-best hypotheses using dialog examples and agenda," in *ACL*, 2008.

[15] M. Li, W. Ruan, X. Liu, L. Soldaini, W. Hamza, and C. Su, "Improving spoken language understanding by exploiting asr n-best hypotheses," *ArXiv*, vol. abs/2001.05284, 2020.

[16] S. Humeau, K. Shuster, M.-A. Lachaux, and J. Weston, "Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring," *arXiv preprint arXiv:1905.01969*, 2019.

[17] L. Logeswaran, M.-W. Chang, K. Lee, K. Toutanova, J. Devlin, and H. Lee, "Zero-shot entity linking by reading entity descriptions," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 3449–3460. [Online]. Available: https://www.aclweb.org/anthology/P19-1335

[18] O. Agarwal and D. M. Bikel, "Entity linking via dual and cross-attention encoders," *arXiv preprint arXiv:2004.03555*, 2020.

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

[20] L. Boytsov and B. Naidan, "Engineering efficient and effective non-metric space library," in *Similarity Search and Applications - 6th International Conference, SISAP 2013, A Coruña, Spain, October 2-4, 2013, Proceedings*, ser. Lecture Notes in Computer Science, N. R. Brisaboa, O. Pedreira, and P. Zezula, Eds., vol. 8199. Springer, 2013, pp. 280–293. [Online]. Available: https://doi.org/10.1007/978-3-642-41062-8_28

[21] J. Lin, R. Nogueira, and A. Yates, "Pretrained transformers for text ranking: Bert and beyond," *arXiv preprint arXiv:2010.06467*, 2020.

[22] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *ArXiv*, vol. abs/1508.04025, 2015.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[24] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8, 1966, pp. 707–710.

[25] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *EMNLP*, 2018.

[26] K. Järvelin and J. Kekäläinen, "Ir evaluation methods for retrieving highly relevant documents," in *ACM SIGIR Forum*, vol. 51, no. 2. ACM New York, NY, USA, 2017, pp. 243–250.