LISTEN, KNOW AND SPELL: KNOWLEDGE-INFUSED SUBWORD MODELING FOR IMPROVING ASR PERFORMANCE OF OOV NAMED ENTITIES

Nilaksh Das[1]* *Monica Sunkara*[2] *Dhanush Bekal*[2]
Duen Horng Chau[1] *Sravan Bodapati*[2] *Katrin Kirchhoff*[2]

[1]Georgia Institute of Technology, USA
[2]Amazon AWS AI, USA

ABSTRACT

Automatic speech recognition (ASR) is increasingly being used in specialized domains such as medical ASR and news transcription. Owing to the lack of high quality annotated speech data in such domains, off-the-shelf models are commonly employed by fine-tuning on domain-specific data. This poses a significant challenge in transcribing long-tail expressions and out-of-vocabulary (OOV) named entities. On the other hand, readily available knowledge graphs (KGs) provide semantically structured knowledge for such domain-specific named entities. In this work, we propose the Knowledge-Infused Subword Model (KISM), a novel technique for incorporating semantic context from KGs into the ASR pipeline for improving the performance of OOV named entities. Our experiments show that KISM improves OOV recall of an ASR model by 4.58% (absolute) for named entities that were not seen during training.

Index Terms— ASR, knowledge graphs, beam search

1. INTRODUCTION

Advancements in automatic speech recognition (ASR) technologies have led to a mass adoption in domain-specific use cases such as medical ASR and news transcription [1, 2]. Such use cases typically include speech heavily consisting of factual context in the form of named entities. However, most auxiliary modeling techniques that augment the acoustic model in the ASR pipeline (e.g., language models) focus only on lexical and syntactic coherence of the output transcription. In order to preserve general semantics, such techniques may introduce evident factual errors, especially in the case of out-of-vocabulary (OOV) entities [3, 4]. For example, a language model (LM) could output a higher score for the transcription *"new daly is the capital of india"*, compared to the factually correct transcription *"new delhi is the capital of india"*, if it has never seen *"new delhi"* during training. On the other hand, readily available knowledge graphs (KGs) provide a tremendous amount of factual information for such entities in a structured format. Thus, KGs can enable an ASR model to become *knowledge-aware* and avoid making such mistakes. To this end, we propose a novel model-agnostic approach to incorporate such structured knowledge into the ASR pipeline for improving performance on OOV named entities (Fig. 1).

In this work, we focus on autoregressive transformer models which have shown superior performance in the ASR domain [5, 6]. We propose the *knowledge-infused subword modeling* approach comprising of a *Knowledge-driven Fuzzy Refinement* technique and a ***K**nowledge-**I**nfused **S**ubword **M**odel* (KISM). The KISM model (redundant acronym used hereon for brevity) works with an autoregressive decoder during beam search to increase the likelihood of decoding named entities that are associated with the knowledge context extracted from a given speech utterance. One major challenge in extracting such knowledge context arises from the unidirectional decoding style (from left to right) employed in the autoregressive inference paradigm, which makes it difficult to detect the complete knowledge context until the entire speech transcription has been decoded. In our previous example, it is not possible to infer that the knowledge context corresponds to *"india"* until the last word has been decoded. We overcome this challenge by employing a two-pass approach, wherein we use the first pass through the ASR model to obtain an initial hypothesis. This hypothesis is used to extract the knowledge context by leveraging an exhaustive knowledge graph (Section 3.1). We then preprocess the knowledge context using our proposed fuzzy refinement approach to identify named entities that have a high fuzzy match with misspelled n-grams in the original hypothesis (Section 3.2). Using these named entities, we then construct the KISM model on-the-fly. Finally, KISM augments the beam search stage in a second pass to increase the likelihood of replacing the misspelled n-grams with the matching entities (Section 3.3).

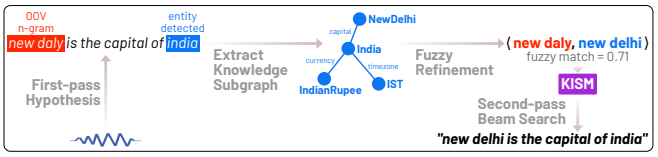

Fig. 1: System overview. KISM fixes the transcription for entity *"new delhi"* by leveraging a knowledge graph and fuzzy refinement.

Through this work, we make the following major contributions:

- Our work proposes a novel technique for infusing ASR inference with structured knowledge-driven context that improves the ASR performance for unseen named entities.
- Our approach is model-agnostic, and can be used with any ASR model that supports beam search decoding at the subword level.
- Since the KISM model is constructed on-the-fly by leveraging KGs in a heuristic manner, our approach does not require any retraining of the underlying ASR model. Additionally, this technique also enables the KISM model to be easily updated with evolving knowledge-based facts by simply updating the KG itself.
- In Section 4, we show that our approach improves the OOV recall for unseen named entities by 4.58% (absolute) from the baseline.

We also perform extensive ablation experiments to demonstrate the efficacy of each component of our proposed approach in obtaining the best results for both OOV recall as well as word error rate. To the best of our knowledge, KISM is the first technique to address the challenge of incorporating knowledge graphs into the ASR pipeline in a heuristic and model-agnostic manner.

*Work conducted as an intern at Amazon AWS AI.

2. BACKGROUND

Knowledge graphs (KGs) can provide contextualized regularization in the ASR pipeline for decoding named entities of interest. In this work, we leverage the DBpedia knowledge graph [7] for this purpose. DBpedia is a large and comprehensive database that aims to represent the web of human knowledge in a semantically structured format. DBpedia also provides DBpedia Spotlight [8], a tool that detects and annotates DBpedia resources such as named entities from unstructured text. Given the large size of some KGs, several works have also been proposed that aim to efficiently operationalize search and traversal over large KGs [9, 10, 11]. We use RDFlib [11] to efficiently navigate the DBpedia KG in this work.

Research in leveraging large-scale KGs for improving ASR performance on OOV named entities is still in an early stage. Few works in the current literature [12, 13] come close to addressing our proposed problem statement. In [12], the authors rescore the ASR N-best list by detecting named entities and computing a semantic relatedness score between the entities based on a TransE metric [14]. The authors of [12] do not report any success with their approach. In [13], authors train a weighted finite-state transducer (WFST) on-demand to rescore a hypothesis lattice. However, the approach in [13] only works on domain-specific language templates, whereas our proposed approach is grammar free, and can be used for any domain. It is also unclear if the approach in [13] can be scaled to large-scale KGs. We show the efficacy of our approach using DBpedia, one of the largest domain-agnostic KGs. Moreover, both works [12, 13] require offline training of artifacts such as the TransE model in [12] and the base WFST in [13]. The TransE model in [12] also needs to be re-trained for any changes in the KG. In contrast, our proposed approach does not require training of any additional artifacts and can directly adapt to changes in the underlying KG.

3. APPROACH

3.1. Extracting Knowledge Subgraph

Multiple named entities occurring concurrently in a speech utterance are more likely to be related to each other. We employ this as a prior and determine if a subset of concurrently occurring named entities are correctly detected by the ASR model in the original transcription. For this, we perform greedy decoding using the CTC head of a hybrid CTC-attention model and obtain a first-pass hypothesis $\hat{\mathbf{y}}$. Leveraging the CTC head ensures a low overhead in the initial pass. We then apply DBpedia Spotlight to detect if any entities $\hat{\mathcal{V}} = \{\hat{v}_1, \ldots, \hat{v}_{\hat{K}}\}$ are already present in $\hat{\mathbf{y}}$. We leverage the exhaustive KG provided by DBpedia coupled with these named entities identified by DBpedia Spotlight to determine all one-hop neighbors of entities in $\hat{\mathcal{V}}$. This yields a knowledge subgraph $\bar{\mathcal{G}} = \{\bar{\mathcal{V}}, \bar{\mathcal{E}}\}$ where $\bar{\mathcal{V}} = \{\bar{v}_1, \ldots, \bar{v}_{\bar{K}}\}$ are the named entity nodes of the graph and $\bar{\mathcal{E}}$ is the set of relations between these nodes. Based on our prior discussed above, we predict that misspelled named entities in $\hat{\mathbf{y}}$ may be present in $\bar{\mathcal{V}}$. Note here that $\hat{\mathcal{V}} \subset \bar{\mathcal{V}}$. In Section 3.2, we describe a fuzzy refinement method that prunes these named entities in $\bar{\mathcal{V}}$.

3.2. Knowledge-driven Fuzzy Refinement

Our assumption is that an ASR model that has not specifically seen named entities during training will make spelling errors for utterances of named entities. Hence, we use the Ratcliff/Obershelp pattern matching algorithm [15] to determine if any named entities in $\bar{\mathcal{V}}$ obtained from the knowledge subgraph $\bar{\mathcal{G}}$ can be used to refine any misspelled phrases in the original transcription $\hat{\mathbf{y}}$. The Ratcliff/Obershelp (RO) pattern matching algorithm is a fuzzy string-matching algorithm that takes two strings S_1 and S_2, and yields a score $D_{RO}(S_1, S_2) \in [0, 1]$ where a score of 1 means both strings are a complete match (sample calculation shown in Table 1).

To apply the fuzzy matching algorithm, we first perform a series of standard text normalization steps on the surface form of each named entity in $\bar{\mathcal{V}}$, giving us a set of normalized surface forms for the named entities, which we denote as $\mathcal{Q} = \{q_1, \ldots, q_{\bar{K}}\}$ where q_k is the normalized surface form of $\bar{v}_k$. Next, we use a separate vocabulary model that contains named entities to detect misspelled words in the original transcription $\hat{\mathbf{y}}$. We then enumerate all possible n-grams in $\hat{\mathbf{y}}$ (up to 5-grams) that have at least one misspelled word. This gives us a set of n-grams containing misspelled words that we denote as $\mathcal{P} = \{p_1, \ldots, p_L\}$. Finally, we perform the fuzzy matching over the cartesian product $P \times \mathcal{Q}$ and filter pairs of misspelled n-grams and normalized surface forms that have a minimum fuzzy matching score of a threshold value α. We denote this set of fuzzy refinements as $\mathcal{R} = \Big\{ (p_l, q_k); D_{RO}(p_l, q_k) \geq \alpha \Big\}$.

We denote the normalized surface forms of named entities thus identified in $\mathcal{R}$ as $Q^* = \{q_1^*, \ldots, q_{K^*}^*\}$ where $K^* = |\mathcal{R}|$. In practice, we minimize the number of fuzzy matching operations performed by only considering pairs of (p_l, q_k) where both have the same number of words/grams. We keep only the top 5 q_k for each p_l, as scored by D_{RO}. From cross-validation, we also set $\alpha = 0.6$. Therefore, this step yields us a set of refinements $\mathcal{R}$ that can substitute misspelled n-grams $\mathcal{P}$ with surface forms of named entities Q^*. In Section 3.3, we describe how we integrate this information derived from structured knowledge with the acoustic model.

3.3. Knowledge-Infused Subword Model (KISM)

The Knowledge-Infused Subword Model (KISM) leverages the fuzzy-matched surface forms Q^* in a second pass through the hybrid CTC-attention model to up-weight entity predictions during beam search. KISM is a subword-level LM based on a prefix tree implementation that emits higher scores for subwords that increase the likelihood of decoding named entities identified in Section 3.2.

For a given example, we compute a subword prefix tree on-the-fly using its corresponding entity surface forms from Q^*. Hence, each node M_a in the KISM prefix tree (except the root node M_ϕ) represents a subword w_a. For each edge in the prefix tree, where the parent node is M_a and the child node is M_b, we assign an energy value $\gamma(M_a, M_b)$ equivalent to the number of valid entity surface forms that can be reached by traversing the tree through node M_b.

The beam search algorithm maintains a list of scored hypotheses with the corresponding decoder states for multiple beam search decoders [16]. The CTC head, attention decoder and LMs are examples of beam search decoders. We employ the KISM prefix tree as one such beam search decoder. By leveraging hash maps, our prefix tree traversal implementation has a linear runtime complexity.

At some arbitrary step n during beam search decoding, the KISM decoder state ψ_n^j for the hypothesis $\mathcal{H}_j$ is represented by some node $M_{\psi_n^j}$ in the prefix tree. The initial decoder state ψ_0^j is set to the root node M_ϕ. Let us also denote the running subword history at step n as $\mathbf{y}_{\psi_n^j} = [y_1, \ldots, y_n]$.

While decoding at step $n+1$ using the last decoder state ψ_n^j, we first obtain the updated decoder state as:

$$\psi_{n+1}^j = \begin{cases} M_a & \text{if } M_a \in \textit{children}(M_{\psi_n^j}) \text{ and } w_a = y_n \\ M_\phi & \text{otherwise} \end{cases} \quad (1)$$

Ground truth	*minquan railway station is a station on longhai railway in minquan county shangqiu henan*
Original ASR output	*minquan railway station is a station on longhai railway in minquan county shangchu henan*
Greedy Fuzzy Refinement	*china railway station is a station on donghaixian railway station minquan county xianghua henan*
ASR output with KISM	*minquan railway station is a station on longhai railway in minquan county shangqiu henan*

(a) Comparison of final output for Greedy Fuzzy Refinement and KISM. In this case, KISM is able to correct the ASR output from *"shangchu"* to *"shangqiu"*.

OOV n-gram	Matched subgraph entity	Fuzzy patterns matched	D_{RO}
shangchu henan	*xianghua henan*	*"ang", "hu", " henan"*	0.79
minquan railway	*china railway*	*"in", "a", " railway"*	0.79
longhai railway in	*donghaixian railway station*	*"onghai", " railway", "i", "n"*	0.76
shangchu	*shangqiu*	*"shang", "u"*	0.75

(b) Fuzzy refinement candidates ranked by fuzzy score (D_{RO}).

Sample calculation: D_{RO}(*"shangchu", "shangqiu"*) = 2 × (|*shang*| + |*u*|) / (|*shangchu*| + |*shangqiu*|) = 2 × (5 + 1) / (8 + 8) = 0.75

Table 1: Qualitative example of Greedy Fuzzy Refinement and KISM output for $\alpha = 0.75$ (exact matches with $D_{RO} = 1$ not shown for brevity). Even though fuzzy matching detects the correct refinement ⟨*shangchu* → *shangqiu*⟩ with a score of 0.75, it doesn't get applied by the Greedy Fuzzy Refinement approach in the final output because a higher-order refinement ⟨*shangchu henan* → *xianghua henan*⟩ with a score of 0.79 already replaces *"shangchu"*. Meanwhile, KISM combines fuzzy matches with acoustic decoding for a more robust refinement.

Next, we initialize an energy vector $\mathbf{g} = [g_{w_1}, \ldots, g_{w_D}]$ for all subwords, D being vocabulary size. Given $M_{\psi^j_{n+1}}$ corresponding to the updated decoder state ψ^j_{n+1}, we compute values in $\mathbf{g}$ as:

$$g_{w_b} = \begin{cases} \gamma(M_{\psi^j_{n+1}}, M_b) & \text{if } \exists M_b \in \textit{children}(M_{\psi^j_{n+1}}) \\ \delta \approx 0 & \text{otherwise} \end{cases} \quad (2)$$

where δ is an arbitrarily small value for avoiding division-by-zero errors. Finally, we compute the subword probabilities emitted by the KISM decoder as $p_{\text{KISM}}(w_a|\psi) = g_{w_a} / \sum_b g_{w_b}$.

Therefore, the KISM decoder up-weights the probabilities for subwords that correspond to children nodes in the prefix tree, where the parent node corresponds to the subword that was last decoded in the given hypothesis. KISM tracks the decoded subwords and updates the state accordingly within the prefix tree. When a leaf node is encountered at some state, i.e., a valid named entity has been decoded, the updated state gets reset to the root node M_ϕ.

4. EXPERIMENTS

4.1. Data

Speech utterances used in our experiments include data from Mozilla's Common Voice [17] and TED-LIUM [18] datasets, as well as TTS data. To effectively evaluate our approach, we consider speech examples containing at least 1 pair of related entities. We found that there is a dearth of such speech examples in public datasets. For instance, the TED-LIUM test set has less than 50 samples that mention a pair of related entities. We overcome this challenge by generating TTS data [19] containing named entities using the publicly available Amazon Polly service [20]. First, we leverage the open-sourced Wikidata5M dataset [21] that provides a semantically annotated text corpus of Wikipedia descriptions. We extract ∼19M sentences by splitting the description paragraphs to obtain short utterances. We then use DBpedia Spotlight to annotate entities in each sentence, and drop sentences not having at least 1 pair of entities that are linked in the DBpedia knowledge graph. Finally, we use Amazon Polly to convert ∼2M sentences thus filtered into speech examples. For generating each sentence, we randomly sample from a set of 9 en-US "Neural TTS" speakers in Amazon Polly (5 female, 4 male voices). We separate ∼90% of this TTS data for training and validation, and combine it with all speech examples from the natural voice datasets [17, 18] that have entities identified using DBpedia Spotlight. For our test split, we use the remaining speech examples from the TTS data, ensuring that each example in the TTS test set has at least 1 entity not seen during training. This yields a test split of ∼10k utterances with OOV named entities.

4.2. Baseline

We use a hybrid CTC-attention model [22] based on the conformer architecture [23]. The encoder has 12 conformer layers with 8 attention heads having output size of 512. The decoder has 1 transformer attention block and the model uses 1000 subword units. All our work is implemented in ESPnet [24]. We start with a model that was pretrained on ∼28k hours of privately obtained voice data of various domains such as news media, public talks and call centre recordings. To get the baseline model, we further fine-tune it on our training data (described in Section 4.1). The model is fine-tuned jointly with CTC loss as well as attention decoder loss using the Adam optimizer [25]. For inference, we use a beam size of 100 with no additional LMs. After fine-tuning, the model has a baseline word error rate (WER) of 5.91% on the test set (Table 2). We also use DBpedia Spotlight for computing the recall of named entities not seen during training (denoted as OOV recall). The baseline model has an OOV recall of 64.71% (Table 2). We also show results from the first-pass greedy CTC decoding used to extract the knowledge subgraph. We see that using only the CTC head, the OOV recall is 60.93%. The inclusion of the attention head in the hybrid CTC-attention baseline improves both OOV recall as well as WER. Hence, we use the model in the hybrid setting as our baseline for all experiments.

4.3. Results

Table 2 shows that the KISM model improves the final OOV recall by an absolute 4.58% with an improvement of absolute 0.25% in WER compared to the baseline. We also perform extensive ablation experiments to show efficacy of each stage of the proposed approach.

4.3.1. Greedy Fuzzy Refinement (Ablation with no KISM)

In this ablation, we evaluate the efficacy of the proposed fuzzy refinement approach, which is an intermediate step in constructing the

	CTC	Attention	KISM	Knowledge Subgraph	Fuzzy Refinement	WER	OOV Recall
First-pass greedy CTC	✓	–	–	–	–	6.71	60.93
Baseline (hybrid CTC-attention)	✓	✓	–	–	–	5.91	64.71
Greedy Fuzzy Refinement	✓	✓	–	✓	✓	5.69	67.61
KISM (no Knowledge Subgraph)	✓	✓	✓	–	–	18.91	65.54
KISM (no Fuzzy Refinement)	✓	✓	✓	✓	–	10.66	68.39
KISM (proposed)	✓	✓	✓	✓	✓	**5.66**	**69.29**

Table 2: Results for different approaches and ablations showing average WER (in %) as well as the OOV recall (in %) for the TTS test set.

KISM model. In the original approach, fuzzy refinement yields a set of top-k named entities for each OOV n-gram having a match score D_{RO} above some threshold α. For this ablation, we modify this approach to follow a greedy heuristic: for each OOV n-gram, we only consider the entity with the highest match score above a threshold α and drop the rest. This gives us a set of refinements, each refinement being a pair of OOV n-gram and a matched entity. We then sort the refinements by D_{RO} and iteratively replace each OOV n-gram with its corresponding entity in this ranked order.

Table 1 shows an example of this Greedy Fuzzy Refinement (GFR) approach. Note in this example that the misspelled word *"shangchu"* has 2 refinements *"xianghua henan"* and *"shangqiu"*, the latter being correct. However, since *"xianghua henan"* has a higher score for the OOV 2-gram *"shangchu henan"*, it takes precedence in the greedy approach, and the correct refinement does not get applied in the final output as *"shangchu"* is already replaced by *"xianghua"*. It is clear from this example that combining the refinements with acoustic information can overcome this issue as *"shangchu"* and *"xianghua"* can be easily distinguished in speech. This motivates our application of KISM on top of fuzzy refinement, combining knowledge-induced artifacts with acoustic decoding.

For obtaining the upper bound OOV recall with this ablation, we perform GFR on the hybrid CTC-attention output. As GFR operates at the word level, from cross-validation we find that using a high value of $\alpha = 0.9$ yields the best results. In contrast, cross-validation shows that a lower $\alpha = 0.6$ is better for KISM as it can filter false-positives by leveraging acoustic scores. We find that decreasing α for GFR increases the chances of false-positive surface forms being substituted, following the greedy heuristic. Conversely, increasing α can filter out true positive surface forms from being included in the KISM prefix tree. Table 2 shows that GFR already improves the OOV recall by 2.9% (from 64.71% to 67.61%). However, KISM is able to further improve OOV recall significantly.

4.3.2. Ablation with no knowledge subgraph

Although variations of subword-level prefix tree models similar to the KISM model have been proposed in the literature [26, 27, 28], KISM's ability to improve OOV recall is derived from the fusion of semantic knowledge extraction and fuzzy matching on top of the beam search augmentation. We test this by removing the knowledge subgraph extraction and the consequent fuzzy refinement steps.

In this ablation, instead of filtering related entities from the first-pass hypothesis, we consider all possible named entities in the KG as candidates Q^* for constructing the prefix tree. Since the set of entities in the KG remain constant, it is no longer required to compute the KISM prefix tree on-the-fly. However, the runtime complexity during beam search, which scales linearly with the branching factor of the prefix tree, goes up significantly as we're now considering entities of an exponentially higher magnitude. We see in Table 2 that this ablation approach does improve the baseline OOV recall by 0.83% (from 64.71% to 65.54%). However, there is a severe deterioration of the WER, which increases from 5.91% all the way up to 18.91%. Since there are significantly more entities in this ablation that are unrelated to the semantic context of the utterance, there is a higher number of false-positive changes induced by the ablated KISM model for closely spelled words during beam search, which increases WER. The slight increase in the OOV recall also demonstrates the power of KISM itself in weeding out true-positive entities even in the presence of hundreds of thousands of noisy entities. Comparing these results to our proposed approach also demonstrates the significance of incorporating structured knowledge. For computational practicality, we only consider all entities in the test set ground truth instead of the entire DBpedia KG.

4.3.3. Ablation with no fuzzy refinement

Extracting the knowledge subgraph provides a semantic context of the utterance in the form of related named entities, whereas the fuzzy refinement step more explicitly constrains this context to match with misspelled n-grams in the first-pass hypothesis. For example, without knowledge subgraph, KISM has more than 2.7M surface forms. Using the subgraph without applying fuzzy refinement, KISM can have up to 180k surface forms. Combining knowledge subgraph and fuzzy refinement yields less than 20 surface forms in most cases.

Hence, we test the efficacy of the proposed fuzzy refinement step by directly using entities from the knowledge subgraph without performing fuzzy refinement. We only perform the knowledge subgraph extraction and use surface forms for the subgraph entities as candidates Q^* for the KISM model. This significantly reduces the number of entities compared to Section 4.3.2, but there still may be many spurious entities, especially for high degree nodes in the KG.

Table 2 shows that performing this ablation improves OOV recall by 3.68% but there is also a significant increase in WER of 4.75%. Without fuzzy refinement, KISM is able to correctly detect entities in the utterance from the knowledge subgraph, but still struggles to lower false-positive changes during beam search. In contrast, the proposed approach with fuzzy refinement further improves OOV recall whilst also improving WER. Fuzzy refinement ensures that the KISM model ignores entities not having a strong match with misspelled n-grams, thus reducing the possibility of false-positives.

5. CONCLUSION

In this work, we propose the novel knowledge-infused subword modeling approach that combines semantic knowledge subgraph extraction with fuzzy refinement and beam search. Our proposed approach improves the performance of an ASR model in correctly predicting OOV named entities from speech by an absolute 4.58%. In future work, we aim to leverage this framework for architectural fusion in order to induce knowledge context directly at earlier layers of the model, thus allowing more robust decoding of named entities.

6. REFERENCES

[1] Mythilisharan Pala, Laxminarayana Parayitam, and Venkataramana Appala, "Real-time transcription, keyword spotting, archival and retrieval for telugu TV news using ASR," *International Journal of Speech Technology*, vol. 22, no. 2, pp. 433–439, 2019.

[2] Monica Sunkara, Srikanth Ronanki, Kalpit Dixit, Sravan Bodapati, and Katrin Kirchhoff, "Robust prediction of punctuation and truecasing for medical ASR," in *Proceedings of the First Workshop on Natural Language Processing for Medical Conversations*, 2020, pp. 53–62.

[3] Imran Sheikh, Dominique Fohr, Irina Illina, and Georges Linares, "Modelling semantic context of OOV words in large vocabulary continuous speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 3, pp. 598–610, 2017.

[4] Hemant Yadav, Sreyan Ghosh, Yi Yu, and Rajiv Ratn Shah, "End-to-End Named Entity Recognition from English Speech," in *Proc. Interspeech 2020*, 2020, pp. 4268–4272.

[5] Yi Ren, Jinglin Liu, Xu Tan, Zhou Zhao, Sheng Zhao, and Tie-Yan Liu, "A study of non-autoregressive model for sequence generation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 149–159.

[6] Xingchen Song, Zhiyong Wu, Yiheng Huang, Chao Weng, Dan Su, and Helen Meng, "Non-autoregressive transformer ASR with CTC-enhanced decoder input," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5894–5898.

[7] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives, "Dbpedia: A nucleus for a web of open data," in *The semantic web*, pp. 722–735. Springer, 2007.

[8] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes, "Improving efficiency and accuracy in multilingual entity extraction," in *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*, 2013.

[9] LU Ruqian, FEI Chaoqun, WANG Chuanqing, GAO Shunfeng, QIU Han, Songmao Zhang, and CAO Cungen, "Hape: A programmable big knowledge graph platform," *Information Sciences*, vol. 509, pp. 87–103, 2020.

[10] Aisha Mohamed, Ghadeer Abuoda, Abdurrahman Ghanem, Zoi Kaoudi, and Ashraf Aboulnaga, "RDFFrames: Knowledge graph access for machine learning tools," *The VLDB Journal*, pp. 1–26, 2021.

[11] "RDFLib: A python library for working with RDF," `https://github.com/RDFLib/rdflib`.

[12] Ashwini Jaya Kumar, Camilo Morales, Maria-Esther Vidal, Christoph Schmidt, and Sören Auer, "Use of knowledge graph in rescoring the n-best list in automatic speech recognition," *arXiv preprint arXiv:1705.08018*, 2017.

[13] Mandana Saebi, Ernest Pusateri, Aaksha Meghawat, and Christophe Van Gysel, "A Discriminative Entity-Aware Language Model for Virtual Assistants," in *Proc. Interspeech 2021*, 2021, pp. 2032–2036.

[14] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.

[15] John W Ratcliff and David E Metzener, "Pattern matching: The gestalt approach," *Dr Dobbs Journal*, vol. 13, no. 7, pp. 46, 1988.

[16] Hiroshi Seki, Takaaki Hori, Shinji Watanabe, Niko Moritz, and Jonathan Le Roux, "Vectorized beam search for CTC-attention-based speech recognition.," in *INTERSPEECH*, 2019, pp. 3825–3829.

[17] "Mozilla Common Voice," `https://commonvoice.mozilla.org/en/datasets`.

[18] François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Esteve, "TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation," in *International conference on speech and computer*. Springer, 2018, pp. 198–208.

[19] "Data generated for 'Listen, Know and Spell'," `https://github.com/amazon-research/listen-know-spell-dataset`.

[20] "Amazon Polly," `https://aws.amazon.com/polly`.

[21] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang, "KEPLER: A unified model for knowledge embedding and pre-trained language representation," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 176–194, 2021.

[22] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi, "Hybrid CTC/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[23] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 5036–5040.

[24] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proceedings of Interspeech*, 2018, pp. 2207–2211.

[25] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[26] Takaaki Hori, Shinji Watanabe, and John R Hershey, "Multi-level language modeling and decoding for open vocabulary end-to-end speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 287–293.

[27] Weiran Wang, Guangsen Wang, Aadyot Bhatnagar, Yingbo Zhou, Caiming Xiong, and Richard Socher, "An Investigation of Phone-Based Subword Units for End-to-End Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 1778–1782.

[28] Duc Le, Mahaveer Jain, Gil Keren, Suyoun Kim, Yangyang Shi, Jay Mahadeokar, Julian Chan, Yuan Shangguan, Christian Fuegen, Ozlem Kalinli, Yatharth Saraf, and Michael L. Seltzer, "Contextualized Streaming End-to-End Speech Recognition with Trie-Based Deep Biasing and Shallow Fusion," in *Proc. Interspeech 2021*, 2021, pp. 1772–1776.