ISSN 0002-306X. Proc. of the RA NAS and NPUA Ser. of tech. sc. 2023. V. LXXVI, N4

UDC 004.832 COMPUTER SCIENCE

DOI: 10.53297/0002306X-2023.v76.4-493

D.S. KARAMYAN, G.A. KIRAKOSYAN

KEYWORD-BIASED SPEECH RECOGNITION: A COMPARATIVE STUDY

The ability to recognize unknown or rare words, such as technical terms and names, is crucial for speech recognition technology to accurately comprehend conversations in context. Nonetheless, current end-to-end speech recognition models often have difficulty in recognizing words that are rarely or never seen during training. This paper examines the effectiveness of different keyword biasing methods, as well as their modifications outlined in previous studies, which do not require any modifications to the ASR model.

Keywords: speech recognition, keywords, contextual biasing.

Introduction. Recent years have witnessed a remarkable advancement in the performance of End-to-End (E2E) automatic speech recognition (ASR) systems with the help of deep learning. Attention-based Encoder and Decoder (AED) [1, 2], Connectionist Temporal Classification (CTC) [3] and Recurrent Neural Network Transducers (RNN-T) [4] have played a significant role in this advancement. However, these systems face difficulties in recognition unknown or uncommon words such as person names, location names or technical terminologies that are rarely or never seen during training. The recently introduced *Whisper Large* [2] model is proficient at recognizing terms, names and other commonly used keywords due to its training on vast amounts of data. Nevertheless, using this model in embedded AI applications that require fast inference and have limited memory resources may not be practical. Aside from the computational aspect, it cannot recognize novel words that were not part of the training set. This problem is not only limited to ASR technology, but also applies to humans as it is difficult to understand a conversation full of unknown words. These words often play a significant role in understanding the overall conversation despite their low frequency of occurrence.

Keyword biasing (also known as *contextual ASR* or *contextual biasing*) is a family of methods of guiding an ASR system towards a specified list of keywords and phrases provided along with the audio to be transcribed. Previous research on contextual ASR can be categorized into *deep biasing* [5-8] and *shallow fusion* [9-11] approaches. While deep biasing methods modify ASR training to incorporate

keyword lists as a secondary input to the ASR model, shallow fusion utilizes keyword lists during decoding to bias the output. This paper will mainly focus on shallow fusion techniques that can be effortlessly integrated into an existing ASR model without the need for retraining, particularly in situations where target domain data is scarce or retraining is not feasible.

On-the-fly (OTF) rescoring [9], is, probably, the most frequently used contextual biasing approach in ASR. Initially, this method was used in hybrid ASR models wherein a new weighted finite state transducer (WFST) is combined with the ASR model's WFST representing bias terms. The weights assigned to the bias terms are dynamically modified during inference, hence the name "on the fly". Another contextual biasing implementation is CTC prefix beam search with OTF rescoring [10, 11]. This involves generating several potential translations *beams* while performing beam search and assigning higher scores to the translations that include bias terms. Several papers [10, 12] explore modifications of this technique. In particular, [10] demonstrates the importance of cost subtraction when a false prefix occurs, while [12] suggests an adaptive boosting method that assigns a smaller boosting score to tokens that have relatively lower acoustic confidence. A line of research [11, 13-15] generates alternative spellings for each bias term and then integrates them into the decoding process. In particular, work in [11], proposed a novel method that can predict how ASR may inaccurately recognize the term and subsequently replace it with the correct spelling.

In this paper, we conduct a comparative analysis of various decoding strategies, including CTC prefix beam search with keyword biasing [10], and modifications proposed in literature, including cost subtraction [10], adaptive boosting [12], and alternate spelling prediction [11], in comparison to baseline strategies such as greedy decoding, vanilla beam search, and beam search with language model (LM) [16]. Furthermore, we evaluate the effectiveness of biasing methods on three different datasets with varying biasing lists, demonstrating their benefits and drawbacks. We also analyze the methods' effectiveness on rare and out-of-vocabulary (OOV) keyword groups.

Beam Search Decoding. The main component of an end-to-end automatic speech recognition system is the acoustic model, which is responsible for converting acoustic signals into a sequence of characters or tokens. When trained from a large amount of labelled speech data, the acoustic model can learn to produce readable transcriptions. However, errors often arise on words that rarely or never appear in the training dataset. In practice, this is hard to avoid: training from enough speech data to *hear* all of the words or language constructions we might need to know is impractical [17]. Therefore, many ASR systems employ an

external LM because these models can be easily trained on huge unlabeled text corpora. The LM is used in fusion with beam search decoding to find the best translation candidates. During the decoding phase multiple alternative token sequences or *beams* are generated which are then scored using the acoustic and language models to select the most likely translation sequence. More formally, given the output of the acoustic model for a given audio signal X, we perform a search to find the sequence of tokens $c_1, c_2, \ldots$ that is most probable according to both the acoustic model output and the language model. Specifically, we aim to find a sequence of tokens C that maximizes the combined objective:

$$Q(C) = log\big(P(C|X)\big) + \alpha log\big(P_{lm}(C)\big) + \beta length(C),$$

where $\mathrm{P(C|X)}$ here is a likelihood of token sequence C estimated by the acoustic model and $P_{lm}(C)$ is the one estimated by the LM. $length(C)$ is a function that returns the length of the sequence C. Parameter α specifies the amount of importance to place on the language model, and β is a penalty term to consider the sequence length in the scores. Larger α means more importance on the LM and less importance on the acoustic model. Negative values for beta will give penalty to longer sequences and make the decoder to prefer shorter predictions, while positive values would result in longer candidates. The objective is maximized using a prefix beam search algorithm proposed by [16].

Prefix Trie. To bias the decoding algorithm towards particular keywords, we first create a prefix tree (trie) for a given list of keywords. A prefix trie is a data structure that allows an efficient search for a specific prefix within a set of keywords. Later, we will use constructed prefix trie in beam search decoding. Fig. shows an example of a prefix trie for a keyword set *palo,paulo,pete*. Each node in the trie has an associated token which can be either a character or a word piece. The first node represents the *root node*, and the colored nodes are *leaf nodes*. The edges of the tree are weighted and indicate the importance of the token as it extends the path within the tree. It is important to note that the outgoing edge from the root node has a weight of 0, indicating that the first token in the keyword will not be taken into account during the decoding process. To simplify a problem, we limit our analysis to single-word keywords, even though keywords can be made up of more than one word.

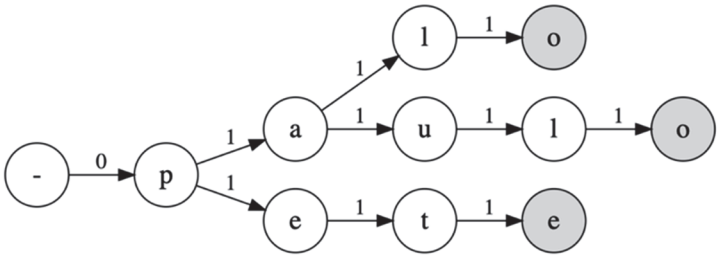

Fig.1. Prefix trie when a biasing list consists of {"palo", "paulo", "pete"}

Keyword-Biased Beam Search. Once a prefix trie of keywords $\mathcal{K}$ is constructed, we can decode the acoustic model output with keyword-biased beam search algorithm. While performing a beam search, the decoding algorithm favours the given keywords if the input speech is pronounced similarly to the given keywords.

To give more importance to the tokens that lead to the next node in the keyword trie, we modify the scoring function $Q(C)$ by adding an extra term:

$$Q_{biased}(C) = Q(C) + \gamma T(C),$$

where γ controls the strength of boosting algorithm and $T(C)$ is a keyword boosting score for beam candidate C:

$$T(C) = \begin{cases} 1, if\ lastWord(C) \in \mathcal{K}, \\ 0, if\ lastWord(C) \notin \mathcal{K}. \end{cases}$$

Here, $lastWord(C)$ is a function that returns the last word of candidate beam C. A high value of γ may lead to overboost (boosting a word as a keyword even if it is not actually presented in speech) and a low value may result in nothing but a vanilla beam search. Moreover, if the prefix consists of only one token, the boosting score is not applied. This is because boosting keywords from the beginning may also lead to overboosting, as one-token prefixes are quite common in candidate beams, particularly when there are a large number of keywords.

Adaptive boosting. To reduce false positives and limit the overboosting effect, an adaptive boosting method is proposed in [12]. More specifically, the proposed method assigns a smaller boosting score to tokens that have relatively lower acoustic confidence, thereby reducing the likelihood of false positives. Let $\hat{y}(t, k)$ denote the log-probability distribution by the acoustic model at time t with k representing the token index. The boosting score for each token at time step t is determined dynamically by its difference in log-probability with the most probable token:

$$\delta(t,k) = \sqrt{\max_k\{\hat{y}(t,k)\} - \hat{y}(t,k)},$$

$$T(C) = \begin{cases} \sigma\big(\delta(t,k)\big), if lastWord(C) \in \mathcal{K}, \\ 0, if\ lastWord(C) \notin \mathcal{K}, \end{cases}$$

where $\sigma(x) = 2/(1+e^x)$ represents a scaled inverse sigmoid function.

Cost subtraction. Work [10] proposes a simple technique to address situations where a false prefix occurs. When the prefix on $\mathcal{K}$ is about to break because the next token does not continue the prefix, we have to subtract the accumulated value up to the current node. For example, consider Fig. 1 and let the current prefix path be *pau* and the upcoming token be *e*. In this case, the accumulated boosting score assigned to the prefix *pau* needs to be subtracted from the overall score because the word *paue* is no longer a prefix on $\mathcal{K}$.

Alternate Spelling Prediction. Work [11] presents a novel alternate spelling prediction (ASP) model which can enhance the effectiveness of a keyword-biased beam search algorithm. The ASP model is a text-to-text, transformer-based, encoder-decoder model. It aims to predict how the ASR system might inaccurately recognize a given keyword or term. For example, if the input is the name "Krisp", the ASP model should predict "Crisp" because that is how the ASR system will recognize the term. Using this ASP model, we can also add weight to token sequences associated with the alternate terms the model suggests. These sequences can then be replaced by the original keywords in the ASR output.

We followed the training and inference procedures as described in the original paper. The ASP model's training data is derived from ASR audio-text paired data. First, we run ASR on audio files to obtain the corresponding text outputs. The outputs are then aligned with reference texts, and incorrectly recognized word pairs are extracted and filtered to exclude insertion or deletion errors and only include non-stopword errors. In contrast to the original paper, we also remove error pairs where reference and predicted words are not phonetically similar to maintain the quality of the data. To align predicted and reference texts, we used an open-source tool called *fstalign*[1].

Evaluation Datasets. *Earnings21 benchmark.* For evaluation we use a publicly accessible Earnings21 dataset [18]. The Earnings21 dataset is a 39-hour corpus of earnings calls containing entity-dense speech from the financial sector. The discussions in these recordings contain industry-specific terminology including the names of companies, products and executives. Moreover, the benchmark comes

[1] https://github.com/revdotcom/fstalign

with two predefined keyword lists - *oracle* and *distractor*. The *oracle* list includes terms and phrases found in the Earnings21 reference transcripts, while the *distractor* list includes all biasing terms from the oracle list along with other company names and renowned CEOs that are not present in the Earnings21 dataset. As we limit our analysis to single-word terms, we excluded phrases from both lists reducing their sizes to 272 and 434, respectively.

Meetings dataset. We also evaluated on an in-house Meetings dataset, consisting of 13 meeting recordings with each recording spanning an average of 2169 seconds (7.8 hours in total). The reference transcripts contain a diverse array of terms and names, which are challenging for the ASR system to recognize, like people's names (such as Caruana, Hovhannes), locations (such as Jamaica, Artsakh), companies and software applications (such as Plantronics, Bloomberg, Discord, Figma). We extracted a list of 178 terms from the reference transcripts.

Podcasts dataset. In order to tune the hyperparameters used in the decoding phase, we gathered the second dataset comprising 22 audio recordings. The total duration of the recordings is 10.5 hours and they encompass conversational discourse involving multiple speakers. We collected a biasing list of 128 terms from the reference texts. A hyperparameter search was conducted on this dataset to find the optimal values of α, β and γ parameters for keyword-biased beam search decoding. The best result was achieved for $\alpha = 0.3, \beta = 0.95, \gamma = 2.4$.

Evaluation Metrics. In line with previous works, we report the Word Error Rate (WER) for all experiments which indicates the percentage of words that were incorrectly predicted. Because the number of keywords is quite small when compared to the size of the text corpus, it's possible that effective methods for recognizing these keywords may not have a significant impact on the WER. For that reason, in addition to WER, we report precision, recall and F1-score on keywords. These metrics are more useful in determining how well the ASR system recognizes the terms from the user's perspective. We aim to improve recall without sacrificing precision and WER.

In addition, we calculate these metrics for three different groups of biasing keywords: *All Words*, which includes every non-stopword word that appears in the biasing list; *Rare words*, defined as those that appear less than 150 times in the ASR training data; and *OOV words*, which are words that do not occur in the ASR training data.

Automatic Speech Recognition Model. Our ASR model is based on a non-autoregressive variant of Conformer-CTC architecture [19] which effectively combines convolutional and transformer blocks to model both local and global dependencies of an audio sequence. We use a medium-size pre-trained Conformer

checkpoint[2] that was made available by Nvidia. We further fine-tune the model on an in-house dataset with around 75,000 hours of English speech. The model generates a probability distribution across subword units with a vocabulary size of 128. We use a subword language model [20] for the LM fusion in beam search decoding. We train the language model on the text part of the ASR training dataset. To provide a comparative analysis, we also report the results of the open-source OpenAI Whisper models [2], which were trained on a massive amount of multilingual speech dataset (680,000 hours).

Alternate Spelling Prediction Model. To train the alternate spelling prediction model, we used roughly 1.15 million word pairs that were mistakenly recognized by an ASR system. Furthermore, we removed error pairs in which the phonetic forms of the reference and predicted words had an edit distance greater than 50%. We used the grapheme-to-phoneme (G2P) library[3] to convert words to the corresponding phoneme sequence.

Similar to [11], our ASP model is also based on a transformer encoder-decoder framework. It has two layers in both the encoder and decoder with two attention heads per layer and 400 units per layer resulting in a total of 6.5 million parameters. However, unlike the original paper, the input and the output subword tokenization is the same as the tokenization used for the ASR model.

At inference time we use beam search to produce a 5-best list of alternate spellings for each keyword. We then filter these alternates to remove bad alternates that are likely to introduce false-positive errors. There are two types of alternates that need to be filtered out - poor matches and common words. We remove the poor matches if the log-likelihood of the hypothesized alternate is lower than the best one by the threshold of 1.0. For common words we filter out any suggested alternate that appeared more than a 1000 number of times in the ASR training data or belongs to the list of top 10,000 frequently used English words[4].

To test the accuracy of the ASP model, we measure the BLEU score [21] between the word pieces of the reference and predicted alternates. Fig.2 shows the results of the ASP model on the test set. For comparison, we present the baseline score for an identity system that keeps the input word unchanged. In addition, we report the score obtained by a refined ASP model using beam search. Fig. 3 illustrates examples of alternates that the ASP model produces.

[2]https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/stt_en_conformer_ctc_medium
[3] https://github.com/Kyubyong/g2p
[4] https://github.com/arstgit/high-frequency-vocabulary/blob/master/10k.txt

Model	BLEU
Identity	0.48
Vanilla ASP	0.6
ASP with beam search	0.605

Fig. 2. Performance of the ASP model with and without beam search in comparison to the identity baseline

Keyword	Top1	Top2	Top3
Keywords seen during training			
hashimoto	hashamoto	hashimoto	hashamato
jupyter	jupiter*	jupitor*	jupitter*
kotlin	cotlin	cotlan*	codlin*
pulumi	pulumi	polumi	poulumi
Keywords unseen during training			
farnoosh	farnosh	farnush	farnash
doernenburg	dornenburg	doernenberg	doernenburg
odersky	oderski	odersky	odderski

Fig. 3. Top three alternates generated by the ASP model with ∗ denoting alternates that were filtered out due to low confidence or being a common word

Results

Baseline methods. Table displays the results obtained by applying keyword-biasing methods along with a comparison to the baseline methods that do not employ any biasing techniques. As shown in Table 1, both greedy decoding and vanilla beam search perform poorly at identifying keywords. This is because the acoustic model alone cannot recognize *OOV* and *Rare* words, which were either absent or infrequent during training. In comparison to the vanilla beam search, LM fusion yields significant improvements in both WER and F1 scores across all three datasets. This is because the LM helps to rectify misspelled keywords as illustrated in Table 2.

Table 1

Performance comparison of keyword-biasing methods with relation to baseline methods. The first section showcases the scores of Whisper Large and Small models. The second section displays the baseline scores generated using three decoding techniques: Greedy, Beam Search, and Beam Search with LM fusion, without any keyword-biasing. Lastly, the third section demonstrates the results obtained from implementing different biasing algorithms

	Podcast				Meetings				Earnings21 - oracle			
	R↑	P↑	F1↑	WER↓	R↑	P↑	F1↑	WER↓	R↑	P↑	F1↑	WER↓
Whisper Large	79.36	97.53	87.51	7.81	62.39	97.46	76.08	9.28	85.61	92.43	88.89	9.72
Whisper Small	73.34	96.72	83.42	7.17	55.64	97.02	70.72	9.36	80.23	91.89	85.66	9.83
Greedy decoding	47.03	92.41	62.34	7.65	39.74	96.27	56.26	12.19	70.38	90.80	79.30	14.25
Beam search (width=16)	46.76	92.88	62.20	7.60	40.26	96.52	56.82	12.10	70.54	91.15	79.53	14.12
Beam search with LM (width=16)	55.59	**95.72**	70.33	6.93	43.33	**96.94**	59.89	**10.81**	73.01	**90.90**	**80.98**	**13.03**
Keyword-biased beam search: (1)	74.45	94.73	83.37	6.95	52.39	90.15	66.27	11.14	81.08	80.50	80.79	13.35
(1) + Cost subtraction: (2)	77.17	92.42	84.11	6.86	54.70	85.45	66.7	11.08	83.90	73.91	78.59	13.44
(2) + Adaptive: (3)	73.03	94.96	82.56	6.87	53.16	90.14	66.88	10.97	82.08	78.62	80.32	13.30
(3) + ASP	**78.84**	94.84	**86.10**	**6.83**	**66.67**	91.55	**77.15**	10.86	**84.00**	76.07	79.84	13.35

Table 2

Transcription samples generated by baseline methods

Reference text	so **ni wang** is now the rebecca's assistent
Whisper Large	so **ni wang** is now rebecca's assistent
Greedy decoding	so **kne waang** is now the rebecca's assistent
Beam search	so **kne waang** is now the rebecca's assistent
Beam search with LM fusion	so **knee wang** is now the rebecca's assistent

Additionally, Table includes the results of the Whisper Large and Small models. These models are proficient at recognizing terms, names and other commonly used keywords due to their training on vast amounts of data. However, they still struggle to recognize novel words that were not present in the training set.

Keyword-biasing methods. The use of keyword-biased beam search resulted in a significant boost in recall across all three datasets. Specifically, when compared to the LM-fused beam search, there is an 18.86% absolute improvement in recall in the Podcasts dataset, a 9.06% improvement in the Meetings dataset and an 8.07% improvement in the Earnings21-oracle dataset. Additionally, by using *cost subtraction* approach we can filter out beams that contain false prefixes, which leads to an overall increase in recall of around 2-3% across all three datasets.

As can be seen from Table 1 by imposing the decoder to produce certain keywords, we observed a reduction in precision where the predicted keywords were not actually present in the reference text. By using *adaptive boosting* approach, we were able to increase precision and limit the overboosting effect.

The last row of Table 1 demonstrates the effectiveness of *alternate spelling prediction* approach. Compared to the LM-fused beam search baseline, the ASP approach resulted in an absolute recall improvement of 23.25%, 23.34% and 10.99% across the Podcasts, Meetings and Earnings21-oracle datasets, respectively. As outlined earlier, to prevent false positive errors, the keyword-biased beam search is not applied right from the beginning of the keyword. This means that keywords like "Krisp" may be recognized as "Crisp" without the possibility of correction. By incorporating ASP alternatives into the decoding process, we can anticipate how ASR may inaccurately recognize the term and subsequently replace it with the correct spelling.

Table 3 illustrates both positive and negative examples of ASR transcription with and without keyword biasing. Some errors occurred when the decoder failed to complete a word by skipping the last characters (e.g., *manue*) which can potentially be caused by adaptive boosting. Other errors arose from the beam pruning logic, where candidates with the correct keyword prefix were pruned

during the beam search due to their relatively lower scores. Another source of error comes from the ASR's inability to safely capture the phonetic prefix of the keyword and hence it will never be boosted (e.g., *ubiquitous*).

Table 3

Positive and negative examples of ASR transcription with and without keyword biasing

Positive examples	
gold text	the airline's ceo oscar munoz issued an apology
no biasing	the airlines ceo oscar munyos issued an apology
with biasing	the airlines ceo oscar munoz issued an apology
gold text	i wanted to interview cathy o'neil
no biasing	i wanted to interview kathy o'neill
with biasing	i wanted to interview cathy o'neil
gold text	the team topologies patterns
no biasing	the team apologies patterns
with biasing	the team topologies patterns
gold text	i'm ashock one of your regular co hosts
no biasing	i'm a shock one of your regular cohorts
with biasing	i'm ashock one of your regular cohorts
Negative examples	
gold text	it's evan bottcher here from melbourne
no biasing	it's evan botcher here from melbourne
with biasing	it's even bottcher here from melbourne
gold text	manuel you are talking to that
no biasing	manua you are talking to that
with biasing	manue you are talking to that
gold text	that data is ubiquitous
no biasing	that data is ibiicuous
with biasing	that data is ibiguous
gold text	this architecture as data mesh
no biasing	this architecture as datamish
with biasing	this architecture as data mish

OOV and Rare words: According to Table 4, it is clear that both greedy decoding and LM-fused beam search decoding are not successful at recognizing *OOV words*. In fact, on the podcast and meetings datasets these methods failed to detect any *OOV words* resulting in a 0% F1 score. LM fusion, on the other hand, significantly improves the recall and F1-score of recognizing *Rare words*, as these words are incorporated in the LM lexicon. By introducing keyword biasing techniques in the decoding process, we were able to drastically increase the recognition accuracy of *OOV* and *Rare words* (see the last row of Table 4).

Table 4

Evaluation results of OOV and Rare words with the numbers x/y representing the Recall and F1 scores, respectively. The third row shows the proportion of Rare and OOV words in the biasing lists for each evaluation dataset

	Podcast			Meetings			Earnings21 - oracle		
	OOV	**Rare**	**All**	**OOV**	**Rare**	**All**	**OOV**	**Rare**	**All**
Number of keywords	12 (9.37%)	49 (38.28%)	128 (100%)	17 (9.55%)	55 (30.9%)	178 (100%)	20 (7.4%)	100 (36.7%)	272 (100%)
Greedy decoding	7.41/13.79	24.37/38.67	47.03/62.34	0.0/0.0	4.51/8.53	39.74/56.26	5.65/10.69	22.42/34.59	70.38/79.30
Beam Search with LM	0.0/0.0	37.39/53.61	55.59/70.33	1.30/2.56	4.51/8.55	43.33/59.89	1.61/3.17	31.03/45.66	73.01/**80.98**
Keyword-biasing + Cost subtraction + Adaptive + ASP	**44.44/60.0**	**72.46/81.62**	**78.84/86.10**	**19.48/32.61**	**51.88/66.13**	**66.67/77.15**	**51.61/67.37**	**57.40/62.15**	**84.00**/79.84

Conclusions. In conclusion, we performed a comparative analysis of various adaptations to the beam search algorithm, including keyword-biasing, LM fusion, cost subtraction, adaptive boosting and alternate spelling prediction. Our evaluation was carried out on three datasets, and the results demonstrate that LM fusion can consistently boost the recall and precision of rare and common words, but does not help in recognizing OOV words. We observed significant improvement in recall when using keyword-biased beam search along with cost subtraction across all three datasets, while the use of the adaptive boosting method mostly improved precision. A further enhancement is achieved by employing an alternate spelling prediction approach.

REFERENCES

1. **Chan W., Jaitly N., Le Q., and Vinyals O.** Listen, attend and spell: A neural network for large vocabulary conversational speech recognition // IEEE international conference on acoustics, speech and signal processing (ICASSP). -2016. -P. 4960–4964.
2. Robust speech recognition via large-scale weak supervision / **A. Radford, J.W. Kim, T. Xu, G. Brockman, et al**: ArXiv preprint arXiv:2212.04356, 2022.
3. **Graves A., Fernández S., Gomez F., and Schmidhuber J.** Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks // Proceedings of the 23rd International Conference on Machine Learning. -2006. -P. 369–376.
4. **Graves A.** Sequence transduction with recurrent neural networks: ArXiv preprint arXiv:1211.3711, 2012.
5. Deep context: end-to-end contextual speech recognition/ **G. Pundak, T.N. Sainath, R. Prabhavalkar, et al** // IEEE spoken language technology workshop (SLT). -2018. -P. 418–425.
6. **Chen Z., Jain M., Wang Y., Seltzer M.L., and Fuegen C.** Joint Grapheme and Phoneme Embeddings for Contextual End-to-End ASR // Interspeech. -2019. -P. 3490–3494.

7. **Alon U., Pundak G., and Sainath T.N.** Contextual speech recognition with difficult negative training examples // ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). -2019. -P. 6440–6444.
8. Contextual RNN-T for open domain ASR / **Jain M., Keren G., Mahadeokar J., Zweig G., et. al**: ArXiv preprint arXiv:2006.03411. -2020.
9. Composition-based on-the-fly rescoring for salient n-gram biasing/ **Hall K., Cho E., Allauzen C., Beaufays F., et. al** // Interspeech 2015, International Speech Communications Association.-2015.
10. **Jung N., Kim G., and Chung J.S.** Spell my name: keyword boosted speech recognition // IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). -2022. -P. 6642–6646.
11. **Fox J. D. and Delworth N.** Improving Contextual Recognition of Rare Words with an Alternate Spelling Prediction Model // Interspeech. -2022.
12. Personalization of ctc speech recognition models/ **Dingliwal S., Sunkara M., Ronanki S., Farris J., et. al** // IEEE Spoken Language Technology Workshop (SLT). -2023. -P. 302–309.
13. **Le D., Koehler T., Fuegen C., and Seltzer M.L.** G2G: TTS-driven pronunciation learning for graphemic hybrid ASR // IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). -2020. -P. 6869–6873.
14. Deep shallow fusion for RNN-T personalization/ **D. Le, G. Keren, J. Chan, J. Mahadeokar, et. al** // IEEE Spoken Language Technology Workshop (SLT).-2021.- P. 251–257.
15. Contextualized streaming end-to-end speech recognition with trie-based deep biasing and shallow fusion/ **Le, Duc, Mahaveer Jain, Gil Keren, Suyoun Kim, et al**: ArXiv preprint arXiv:2104.02194. -2021.
16. **Hannun A.Y., Maas A.L., Jurafsky D., and Ng A.Y.** First-pass large vocabulary continuous speech recognition using bi-directional recurrent dnns.: ArXiv preprint arXiv:1408.2873, -2014.
17. Deep speech: Scaling up end-to-end speech recognition/ **Hannun A., et al**: ArXiv preprint arXiv:1412.5567.-2014.
18. Earnings-21: A Practical Benchmark for ASR in the Wild/ **Rio M. D., et al.** -2021.
19. Conformer: Convolution-augmented transformer for speech recognition/ **Gulati A., et al.** ArXiv preprint arXiv:2005.08100. -2020.
20. **Heafield K.** KenLM: Faster and smaller language model queries // Proceedings of the sixth workshop on statistical machine translation. -2011. -P. 187–197.
21. **Papineni K., Roukos S., Ward T., and Zhu W.-J.** Bleu: a method for automatic evaluation of machine translation // Proceedings of the 40th annual meeting of the Association for Computational Linguistics. -2002. -P. 311–318.

National Polytechnic University of Armenia. The materials is received on 11.01.2024.

Դ.Ս. ՔԱՐԱՄՅԱՆ, Գ.Ա. ԿԻՐԱԿՈՍՅԱՆ

ԱՌԱՆՑՔԱՅԻՆ ԲԱՌԵՐԻ ՀԻՄՔՈՎ ԽՈՍՔԻ ՃԱՆԱՉՈՒՄ: ՀԱՄԵՄԱՏԱԿԱՆ ՈՒՍՈՒՄՆԱՍԻՐՈՒԹՅՈՒՆ

Անհայտ կամ հազվադեպ գործածվող բառերը, ինչպիսիք են տեխնիկական տերմիններն ու անունները ճանաչելու ունակությունը կարևոր է խոսքի ճանաչման տեխնոլոգիայի դեպքում՝ խոսակցությունները ճշգրիտ ընկալելու համար: Այնուամենայնիվ, ըստ խոսքի ճանաչման ներկայիս մոդելների՝ հաճախ դժվար է ճանաչել այն բառերը, որոնք հազվադեպ են գործածվում կամ երբեք չեն հանդիպում մոդելի ուսուցման ընթացքում: Աշխատանքում ուսումնասիրվել է առանցքային բառերի կանխակալման տարբեր մեթոդների արդյունավետությունը, որոնք ուրվագծվել են նախորդ հետազոտություններում և չեն պահանջում որևէ փոփոխություն խոսքի ճանաչման մոդելում:

Առանցքային բառեր. խոսքի ճանաչում, առանցքային բառեր, կոնտեքստային կանխակալություն:

Д.С. КАРАМЯН, Г.А. КИРАКОСЯН

РАСПОЗНАВАНИЕ РЕЧИ НА ОСНОВЕ КЛЮЧЕВЫХ СЛОВ: СРАВНИТЕЛЬНОЕ ИССЛЕДОВАНИЕ

Способность распознавания неизвестных или редких слов, таких как технические термины и имена, является ключевой для того, чтобы технология распознавания речи точно понимала разговоры в контексте. Тем не менее, текущим моделям распознавания речи часто бывает сложно распознавать слова, которые редко или вообще не встречаются во время обучения. В данной работе изучается эффективность различных методов смещения ключевых слов, а также их модификации, которые были описаны в предыдущих исследованиях и не требовали внесения изменений в модель распознавания речи.

Ключевые слова: распознавание речи, ключевые слова, контекстуальная предвзятость.