NAVER KAIST

Spell my name: keyword boosted speech recognition

Namkyu Jung[1], Geonmin Kim[1], Joon Son Chung[2]

[1]Naver Corporation, South Korea,
[2]Korea Advanced Institute of Science and Technology, South Korea

Session: SPE-22: Adaptation and Personalization for Speech Models

Contents

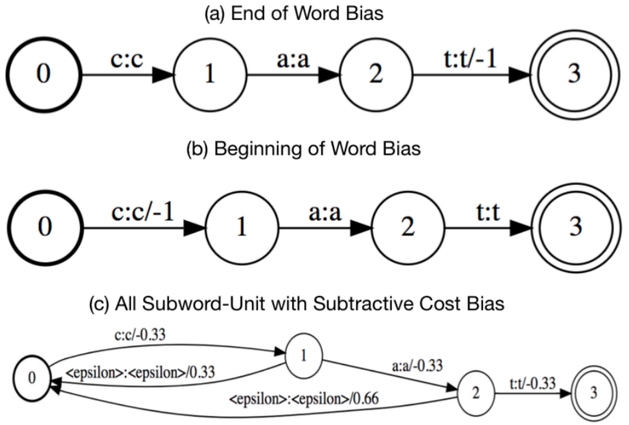
Background

- **Keyword**: uncommon words play an important role in the meaning of the text (person name, technical terminology).
- Recognition of keywords is hard but important to understand conversations in context.
- **Contextual biasing** to a specific domain is essential for production-level ASR.
- Hard to use keywords in modeling directly
  - Impossible to prepare for every usage environment.
  - Hard to know in advance.
  - Sensitive with the privacy issue.

Related Works

Biasing FST

- Hall, K. et al.[1] and P Aleksic et al.[2] try a composition with WFST and n-gram to bias.
- Duc Le et al.[3] shows deep shallow fusion between RNN-T and deep personalized LM.
- Duc Le et al.[4] presents trie-based deep biasing to use open-domain biasing list with RNN-T and NNLM.



These methods need an external language model to bias and WFST to be decoded with.

Moreover, they require additional training processes.

[1] Hall, Keith, et al. "Composition-based on-the-fly rescoring for salient n-gram biasing." (2015).
[2] Aleksic, Petar, et al. "Bringing contextual information to google speech recognition." (2015).
[3] Le, Duc, et al. "Deep shallow fusion for RNN-T personalization." *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021.
[4] Le, Duc, et al. "Contextualized streaming end-to-end speech recognition with trie-based deep biasing and shallow fusion." *arXiv preprint arXiv:2104.02194* (2021).

Related Works

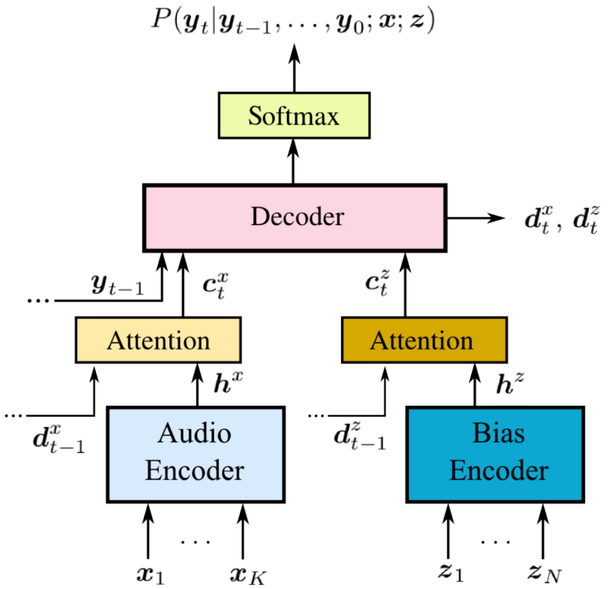
Biasing within the model

- G Pundak et al.[1] and U Alon et al.[2] presents Contextual-LAS (CLAS) to bias model in all-neural way
- M Jain, et al.[3] shows Contextual RNN-T on an open domain video ASR task.

These models can be used only with the ASR model trained together.

It is also difficult to control the strength of the bias since it is trained in end-to-end manner.

Moreover, they also require additional training processes.



[1] Pundak, Golan, et al. "Deep context: end-to-end contextual speech recognition." *2018 IEEE spoken language technology workshop (SLT)*. IEEE, 2018.
[2] Alon, Uri, Golan Pundak, and Tara N. Sainath. "Contextual speech recognition with difficult negative training examples." *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019.
[3] Jain, Mahaveer, et al. "Contextual RNN-T for open domain ASR." *arXiv preprint arXiv:2006.03411* (2020).

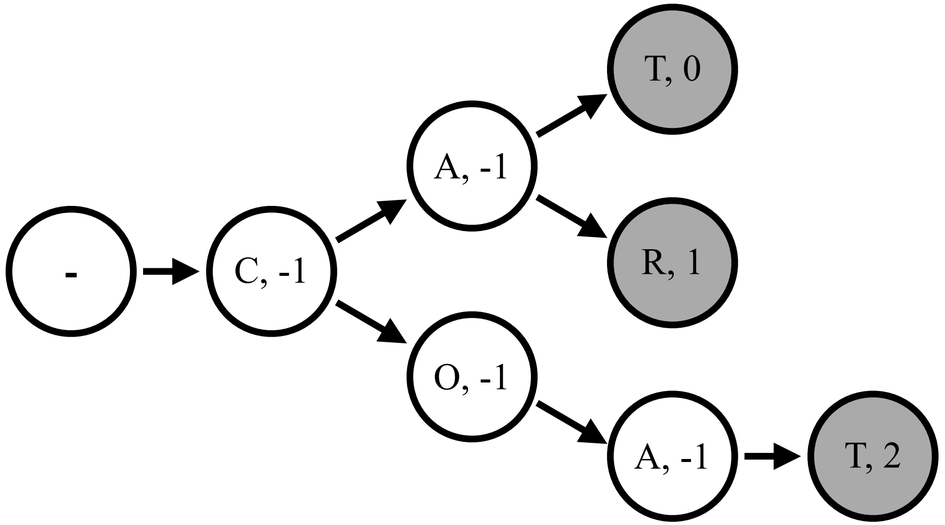
Our Contributions

- We propose the keyword-boosted speech recognition for given keyword list
  - Without any language models and WFST.
  - Without any additional training processes.
  - Independent of the type of ASR model as long as it uses the beam search decoder.

Main Idea

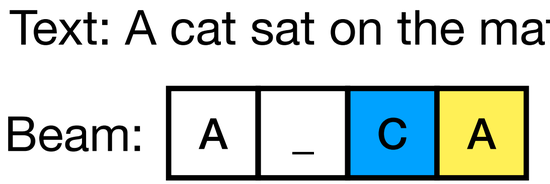
Keyword Prefix Tree

- Make prefix tree (trie) with a keyword list for the keyword search.
- Each node in the tree has a token and an index.
  - A token is a character or a subword token that makes keyword.
  - If the path from the root node constitutes a keyword, node index is the index of the keyword in the list.
  - Otherwise, the node index is a non-keyword index (-1).

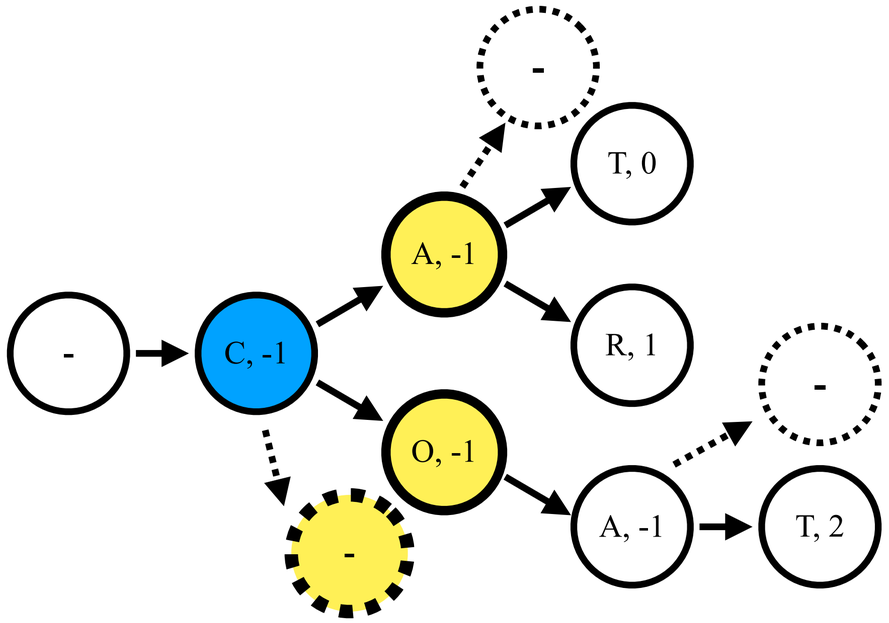


Main Idea

Keyword-boosted Beam Search



- For each beam search step, we take two steps:
- **Step 1** Update tree node: Update the tree node as the state of the beam has changed.
  - The node proceeds to the next node if there is a child node having token same as the changed state.
  - If there isn't, the tree node goes back to the root node.
    - If the state is one of the children of the root node, update the node with it since it can be the beginning of the new keyword path.



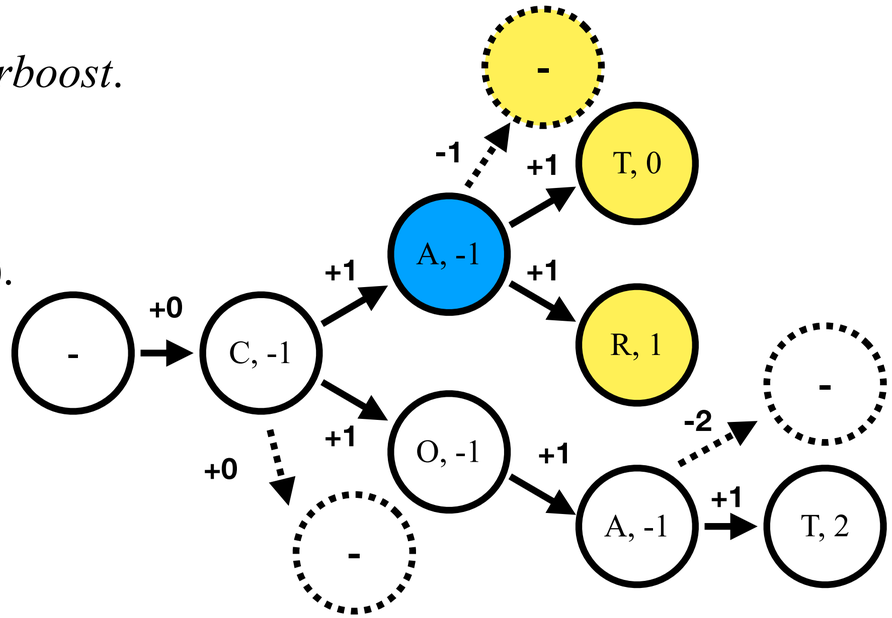
Main Idea

Keyword-boosted Beam Search

Text: A cat sat on the mat

Beam: | A | _ | C | A |

- **Step 2** Calculate keyword score: As the node has changed, we update the keyword score table $K_{t,b} \in \mathbb{R}^V$.
  - If the node is the root node, $K_{t,b}(s) = 0, \forall s \in V$.
    - Don't assign any score from the beginning of the keyword to alleviate *overboost*.
  - If the node is not the end of a keyword (node index is -1),
    $K_{t,b}(s) = -w_k * \text{depth}_K(n_b), \forall s \in V - \{\text{blank}\}$
    (the blank means the CTC blank, blank state doesn't have any keyword score).
    - Subtract the cumulative score since it fails to complete a keyword (subtractive cost).
    - Do not subtract this on the end of each keyword because it makes the complete keyword path.
  - For states included in the children of the node, add score, i.e. $K_{t,b}(s) = w_k, \forall s \in \text{children}_K(n_b)$

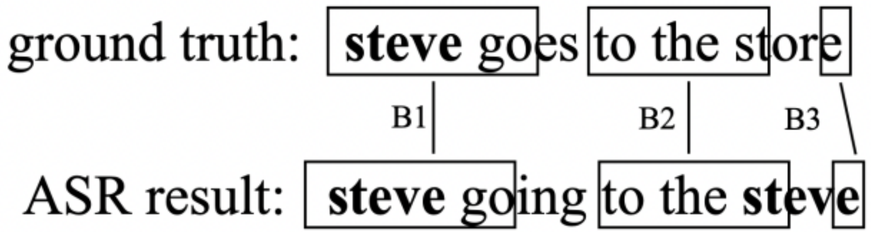


- **Consequently, the beam with a keyword whose length l must be boosted with the keyword score** $(l-1)w_k$

Experiments

Metric

- In order to verify that the keyword occurs at the right place, we align[1] the ground truth and the recognition result.
  - If the keyword is in the same block at the both sides, it counts as a true positive.
  - If the keyword is in the block of the ground truth but not in the corresponding block of the decoded result, it counts as a false negative.
  - If the keyword is in the block of the decoded result though but not in the corresponding block of the ground truth, it counts as a false positive.



TP: 1
FP: 1
FN: 0

Precision: 100%
Recall: 50%
F1-Score: 66.7%

[1] python3 difflib (https://docs.python.org/3/library/difflib.html)

Experiments

Librispeech Dataset

- We extract a list of keywords from LibriSpeech dataset.
  - Each file in LibriSpeech is recorded from a certain book.
  - We collect every transcription in LibriSpeech training set and extract keywords from the collected text by using TF-IDF method.
  - We apply a keyword list consisting of keywords extracted from the same book from which a file of the dev/test dataset of LibriSpeech.
  - We experiment two kinds of keyword list, top 1% score keywords (16.6 words on average) and top 5% score keywords (85.5 words on average) based on TF-IDF score.

# keywords	dev-clean	dev-other	test-clean	test-other
1%	714 (1.3)	1,008 (2.0)	750 (1.4)	1,178 (2.3)
5%	2,632 (4.9)	3,842 (7.5)	2,584 (4.9)	4,126 (7.9)

Table 1. The number of keyword occurrence (the ratio to total word occurrence in %). Keywords account for only a very small percentage of total words.

Experiments

Model setting

- We demonstrate the effect for standard and low-resource environments by using wav2vec 2.0 models trained using 960 hours and 100 hours of data, respectively.
- CTC Beam search is used with beam width = 1500.
- We will compare the case of LM weight w_{LM} = 1.0 or 0.0 (no LM) with n-gram LM since the use of LM is optional in our method.
- w_k (keyword weight) is used as 0 (no boosting) / 0.6 (moderate) / 1.2 (strong).

Experiments

Results

	LM	✗			✓		
	w_k	P	R	F1	P	R	F1
100h	0.0	**98.9**	86.0	92.0	**99.0**	89.5	94.0
fine	0.6	98.9	92.4	95.5	98.9	92.5	95.6
-tuned	1.2	97.5	**94.7**	**96.1**	98.7	**93.2**	**95.9**
960h	0.0	**99.6**	94.5	97.0	**99.3**	95.6	97.4
fine	0.6	99.2	97.7	**98.5**	99.2	97.9	98.5
-tuned	1.2	97.8	**98.5**	98.1	98.8	**98.4**	**98.6**

Table 3. Precision (P), Recall (R) and F1-score (F1) on the LibriSpeech test-clean with boosting 1% keywords for 100h fine-tuned, 960h fine-tuned model respectively.

- Although the precision is slightly decreased, the gain of recall is significant, resulting in a large gain in the F1-score.
- With 100h fine-tuned model, we get a significant improvement of keyword recall from 94.5% to 98.5%.
- Gain is maximized when LM is not used and it shows similar performance as the result with LM.

Experiments

Results

	LM	✗		✓	
	w_k	U-WER	B-WER	U-WER	B-WER
100h	0	2.91	16.78	2.34	11.47
fine	0.6	2.89	9.42	2.34	8.73
-tuned	1.2	2.89	**7.19**	2.33	**7.88**
960h	0	2.08	6.68	1.75	4.62
fine	0.6	2.08	3.08	1.75	2.57
-tuned	1.2	2.09	**2.91**	1.75	**1.88**

Table 3-1. U-WER and B-WER on the LibriSpeech test-clean with boosting 1% keywords for 100h fine-tuned, 960h fine-tuned model respectively.

- U-WER: unbiased WER measured on words NOT IN the biasing list
- B-WER biased WER measured on words IN the biasing list

- The effectiveness can be measured by using the metrics proposed in a previous work. [1]
- U-WER is almost the same since our method hardly changes words other than keywords.
- B-WER improvement is significant.

[1] Le, Duc, et al. "Contextualized streaming end-to-end speech recognition with trie-based deep biasing and shallow fusion." *arXiv preprint arXiv:2104.02194* (2021).

Experiments

In-house Dataset

- Korean-language in-house datasets
  - Clova Note (transcription service) dataset
    - 5,446 audio segments from 365 sessions, 7.5 keywords in average extracted by humans
  - V Live (Live show for celebrities) dataset
    - 17 full audio recordings from K-pop groups (*BTS, Blackpink*)
    - keyword list contains the members' name (real name and stage name) of each group
      - 25 words for *BTS* and 11 words for *Blackpink*.
- We use a wav2vec 2.0 model trained by general-domain Korean language dataset.

Experiments

In-house Dataset

- We use higher w_k because Korean has more letters and shorter words in average.
- CER improvement is small since the keywords do not make up a large proportion of the total word occurrences, but the improvement is consistent.
- Keyword recall has improved from 61.9% to 82.3% in the NAVER VLive dataset.
- This proves that this method can be effectively utilized for real-world ASR services when the context can be specified.

data	w_k	CER	Precision	Recall	F1-score
Clova Note	0	8.07	**98.9**	91.9	95.3
	1	7.92	98.7	93.8	96.2
	3	7.81	98.4	95.5	96.9
	5	**7.78**	98.1	96.3	**97.1**
	7	7.90	97.7	**96.5**	97.1
NAVER VLive	0	16.74	**95.8**	61.9	75.2
	1	16.70	95.6	65.8	78.0
	3	16.62	93.8	74.0	82.7
	5	**16.58**	91.7	79.9	**85.4**
	7	16.60	87.5	**82.3**	84.8

Table 5. Result on *Clova Note* and *VLive* datasets

Experiments

Example from V Live dataset

No Boosting

Boosting with $w_k = 3$

keywords

김남준, 김석진, 김태형, 남준, 민윤기, 박지민, 방탄, 방탄소년단, 뷔, 비티애스, 비티에스, 석진, 슈가, RM, 아미, 윤기, 전정국, 정국, 정호석, 제이홉, 지민, 진, 태형, 호석, BTS, 블랙핑크, 블핑, 김지수, **지수**, 블링크, 로제, 김제니, 채영, 리사, 박채영, 제니

Conclusions

- This method is helpful for recognizing uncommon keywords.
- It can be applied to the beam search decoder naturally without any ASR model dependencies.
- It does not need any further text data or training process.
- The effectiveness is more noticeable in the low-resource environment.
- We can alleviate the overboost by controlling the keyword weight.

Appendix

Librispeech Result

	$w_{LM}=0.0$				$w_{LM}=1.0$			
	dev		test		dev		test	
	clean	other	clean	other	clean	other	clean	other
100h fine-tuned								
No Boosting	3.15	6.42	3.06	6.10	2.41	4.94	2.44	4.80
1% Keywords	3.08/3.06	6.35/6.33	2.96/**2.94**	6.00/5.95	2.39/**2.37**	4.89/4.88	2.41/**2.40**	4.78/**4.76**
5% Keywords	3.06/**3.01**	6.28/**6.27**	2.95/2.96	5.97/**5.91**	2.39/2.38	**4.86**/4.89	2.41/2.42	4.78/4.78
960h fine-tuned								
No Boosting	2.16	4.56	2.13	4.46	1.77	3.51	1.78	3.61
1% Keywords	2.13/2.14	4.49/4.48	2.09/2.10	4.39/4.38	1.74/1.75	3.48/3.48	1.76/1.76	3.59/**3.57**
5% Keywords	**2.09**/2.13	**4.42**/4.47	**2.08**/2.15	**4.37**/4.41	**1.72**/1.75	**3.43**/3.47	**1.75**/1.76	3.60/3.61

Table 2. Word Error Rates (WER) on LibriSpeech with and without n-gram LM and keyword boosting $w_k=(0.6/1.2)$.

- The improvement in WER is relatively small but effective in all scenarios.
- Boosting with 5% of the keywords generally outperforms the other setups
- Optimal keyword weight differs as the number of keywords

Appendix

Librispeech Examples

Positive results	keywords	**milner**,elmwood, sandford, woodley, ojo, dorothy, ozma, scarecrow, miss, lord, tottenhots, pumpkinhead, ...
	ground truth	miss **milner** you shall not leave the house this evening sir
	$w_k = 0$	miss **millner** you shall not leave the house this evening sir
	$w_k = 1.2$	miss **milner** you shall not leave the house this evening sir
Positive results	keywords	cap'n, boolooroo, button, ghip, trot, **pinkies**, ghisizzle, blueskins, calder, bill, marianna, angareb, tiggle, ...
	ground truth	you are not like my people the **pinkies** and there is no place for you in our country
	$w_k = 0$	you are not like my people the **pinkeys** and there is no place for you in our country
	$w_k = 1.2$	you are not like my people the **pinkies** and there is no place for you in our country
Negative results	keywords	servius, praetors, senate, laws, solon, hovel, despotism, julian, decrees, **athens**, edicts, ...
	ground truth	the worthy friend of **athanasius** the worthy antagonist of ...
	$w_k = 0$	the worthy friend of **athanasius** the worthy antagonist of ..
	$w_k = 1.2$	the worthy friend of **athenasius** the worthy antagonist of ...

Table 4. Positive and negative samples of transcription on LibriSpeech with keyword boosting or not.