On the acoustic unit choice for the keyword spotting problem

Aliaksei Kolesau[1,*], Dmitrij Šešok[1]
Department of Information Technologies, Vilnius Gediminas Technical University, Saulėtekio al. 11, LT-10223 Vilnius, Lithuania[1]
aliaksei.kolesau@vilniustech.lt

Abstract: *In this paper we examine the results of using different acoustic units for the building of keyword spotting system. The choice of the acoustic unit greatly influences the quality of the resulting system given the dataset and the model complexity. Decomposing the keyword into simple acoustic units requires the prior knowledge. This knowledge might make the task easier (so the resulting accuracy will be higher), but on the other hand even slightly incorrect priors could mislead the model so the quality might drop significantly. We compare using phonemes, syllables, words and several synthetic acoustic units for Russian language. We show that for modern keyword spotting systems phonemes is a robust and high quality choice, especially in low-resource setting.*
Keywords*: KEYWORD SPOTTING, VOICE ACTIVATION, ACOUSTIC UNIT*

1. Introduction

Voice control is used in a large number of modern mobile and embedded devices. Nevertheless the speech recognition of the whole audio stream is not possible because of both privacy and resource consumption reasons. In order to initiate a voice control most devices use a voice activation system. Its task is to find the specified keyword (e.g. “Alexa”) or keyphrase (“Ok, Google”) in the audio stream and pass the control for more high-level system. Because of the embedded usage keyword spotting system must have a high accuracy, work in real-time and consume a small amount of CPU and RAM.

The keyword spotting task has been attracting both research [1], [2] and industry [3], [4] for several decades. Since the task of formulating an algorithm for determining whether a code phrase has been uttered in an audio stream is difficult to formulate, it is not surprising that heuristic algorithms and machine learning methods have long been used for the voice activation problem [5].

The requirements of working in real-time fashion and consuming a small amount of CPU make a restriction on the size of the keyword spotting model. This in turn limits the complexity and the modeling power of the model. In order to solve the hard problem of finding the keyphrase we can decompose the problem into detecting simpler acoustic events such as syllables or phonemes. The linguistic prior knowledge of how the keyword is decomposed in the smaller parts might greatly improve the quality of the system and make it more interpretable. On the other hand such decompositions are only the approximations of the real world. Building end-to-end model might be more profitable when a large dataset and a big model are available.

This paper examines using the phonemes, syllables, words, uniform and adaptive splitting of the words as target acoustic events. Also we investigate the question of using as targets the events that happen only inside the keyword or all the events of the given type (e.g. phoneme “g” happens not only in “Google”, but also, for example, in ”great” – should we use it as a target during the learning of the model?).

2. Typical voice activation system

Most of modern voice activation systems consist of the following parts [5]:

- feature extraction,
- acoustic model,
- decoding.

Feature extraction is the part of converting the source audio stream into acoustic features. Usually it is done by segmenting the audio and computing some signal processing transformation for each segment (frame) separately. In this paper we compute 80 log-mel filterbanks for the frames of 25 ms length and 10 ms offset.

Acoustic model is a system that generally computes the probability of acoustic observations, which often comes down to computing $P(u \mid O)$, where u is an acoustic unit and O are acoustic observations).

The decoding is the process of determining the state sequence with the reference to acoustic observation and acoustic model in order to determine whether a keyword has been uttered or not.

We follow the example of [3], where authors apply an acoustic model in a form of deep neural network to extracted log-mel filterbanks (feature extraction) and decide whether the keyword was uttered by aggregating deep neural network outputs and comparing them with a threshold (decoding). See Fig. 1 for the illustration.

The aggregation of the acoustic model outputs consists of the two following steps.

Raw posteriors from the neural network are noisy, so we smooth the posteriors over a fixed time window of size W (frames). Let's denote by $p_{f,i}$ the i-th output of the neural network at the frame f. Then the smoothed output $p'_{f,i}$ is computed by:

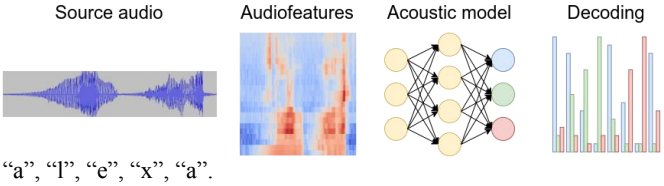
$$p'_{f,i} = \frac{1}{f - H + 1} \sum_{k=H}^{f} p_{k,i}, \qquad (1)$$

where $H = \max\{1, f - W + 1\}$ is the index of the first frame within the smoothing window.

The probability p_f of the keyphrase utterance ending in the frame f is computed with a sliding window of the length S frames, as follows:

$$p_f = \max_{F \le f_1 < f_2 < \ldots < f_n = f} \sqrt[n]{\prod_{i=1}^{n} p'_{f_i,i}}, \qquad (2)$$

where $F = \max\{1, f - H + 1\}$ is the index of the first frame within the sliding window, f_i is the index of the frame where the i-th acoustic event has happened according to the model, n is the number of consecutive acoustic events that must happen in order for the whole keyphrase to happen. For example, in order for the work “Alexa” to happen, five following phoneme events must happen:



“a”, “l”, “e”, “x”, “a”.

Fig. 1 In the first stage we compute log-mel filterbanks for the source audio. Next we apply neural network. Each output gives the probability of the the specific acoustic event happening in the current frame. Finally we aggregate all these probabilities to get the resulting probability of keyphrase utterance.

3. Acoustic units

In this section we describe all the acoustic unit types we used in our experiments.

3.1 Phonemes

In phonology, a phoneme is a unit of sound that dis tinguishes one word from another in a particular language [6]. For example, in English the words “sin” and “sing” are distinguishable by the phoneme “g”. International Phonetic Alphabet is the example of phonetic alphabet suitable for different languages.

Each phoneme of the target keyword can occur not only in the keyword itself, but in the other words. There is a choice of how to handle these occurrences during the training. We denote by “own phonemes” the variant when we optimize our neural network to use only the phonemes in the keyword as targets (and handle other cases as fillers) and by “all phonemes” the case when we optimize the neural network to find the phonemes regardless of whether they are the part of keyword or other word.

“Own phonemes” choice encourages the model to learn the contextual information. It might improve the accuracy of voice activation system, but at the same time the real targets are not phonemes anymore, which may “confuse” the model and thetraining may fail.

3.2 Syllables

A syllable is a unit of organization for a sequence of speech sounds [7]. It is typically made of a syllable nucleus (most often a vowel) with optional initial and final margins (typically, consonants). For this paper we consider the syllable as a sequence of phonemes.

As with phonemes, we have the choices of “own syllables” and “all syllables”. The number of syllables per word is almost always less than the number of phonemes per word, so this choice makes a neural network and a decoding a bit simpler, but the duration of syllable is more variable so the distinction task can be more complex.

3.3 Words

As a baseline we can use the whole word as a target. For example we need two neural network outputs for one word keyphrases: the filler probability and the word probability itself.

3.4 Phoneme transitions targets

We tried to use the transition between phonemes as our targets. It reduces the number of targets by one comparing to phonemes. We also distinguish “all” and “own” scenario for this target type.

3.5 Uniform and adaptive targets

The decomposition of words into phonemes is performed with some assumptions based on the linguistic knowledge. It can be argued, that as an every model it has its errors, so it might be profitable to use simple decomposition and let a neural network to deduce actual patterns from data. In order to check this hypothesis we propose two methods of decomposing words into subword targets.

First, we split word into *n* parts uniformly by the duration of the word. We call this option “uniform targets”. *n* is chosen via hyperparameter search.

The phonemes are irregular in duration. That’s why uniform splitting might result in very different types: several phonemes, one phoneme, transition between phonemes. Therefore we propose the second way of splitting the word: “adaptive targets”. Here we split the word in n parts minimizing the sum of distances from the frame to the average frame in the split in log-mel filterbank space. This can be done effectively via dynamic programming. As for “uniform target” we choose *n* via hypeparameter search.

4. Experiments

4.1 Dataset

For our experiments we used our Russian in-house private dataset of 174619 audio files, 139516 were used for training, 17820 for validation and 17283 for testing. We got the phoneme targets via forced alignment with our speech recognition model. The keywords we used for our experiments are shown in Table 1.

Table 1: *The keywords used for our experiments.*

Word	Phonemes	Syllables	Positive samples in the testset
“Алиса” (Alice)	5	3	7455
“Включи” (turn on)	6	2	542
“Тебя” (you)	4	2	462
“Мне” (me)	3	1	889

4.2 Model

We used a time-delay neural network (TDNN) [8] with ReLU non-linearity. The layers specifications are presented in Table 2. The whole receptive field at frame *f* is made up from the frame *f −28* to the frame *f +12*. The number of units in the last layer is determined by the chosen targets. For example, if we train model to distinguish phonemes of the word “Alice” we need 6 outputs: one for the filler and 5 for the phonemes.

Table 2: *Specifications of the used neural network.*

Layer	Units	Context
1	192	[-8, 1, 2]
2	96	[-3, 2]
3	192	[-8, 1, 2]
4	96	[-3, 2]
5	192	[-3, 2]
6	96	[-3, 2]
7	-	[0]

4.3 Training

We train all our models with SGD optimizer for per-frame cross-entropy loss. The learning rate and batch size were chosen via random search. The training is performed for 10 epochs. If the validation loss between consecutive epochs differs by less than 0.001 the learning rate is decayed by the factor of 4/3.

5. Results and discussion

25 runs were made for each keyword and each target. We tune the threshold in such a way that false reject rate on the test set is less than 0.1 and the false alarm rate is minimized. The numbers of positive samples are very different for different words, so we report the relative drop of the quality comparing to the best option for the given word. The results are in Table 3. The best options are highlighted in bold. The value *n* is specified in the brackets, when appropriate.

Table 3: *Drop of the best false alarm rate comparing to the best choice.*

	Алиса	Включи	Тебя	Мне
Own phonemes	5%	1%		54%
All phonemes	36%	**0%**	**0%**	**0%**
Own syllables	17%	150%	114%	141%
All syllables	34%	131%	105%	137%
Own phoneme transitions	49%	24%	123%	88%
All phoneme transitions	22%	53%	91%	102%
Uniform target	**0% (6)**	19% (6)	95% (3)	35% (7)
Adaptivet arget	10% (5)	28% (8)	152% (2)	54% (7)
Word target	80%	733%	176%	139%

We suggest the following points from experiments:

• phonemes are a great choice for all cases, especially when the number of positive samples is small,

• word target gives very bad accuracy comparing to the phonemes,

• when the number of positive samples is big all the options work reasonably, but when the number is small some options are clearly better than other,

• adaptive splitting doesn't give any advantage comparing to uniform splitting.

Our experiments show that phonemes could be used as a baseline for a keyword spotting system. It can be beaten if the dataset is big, but still it shows comparable results. Also it can be seen that "own" options is preferable when the dataset is large, so the model can take advantage of the contextual information. When the dataset is small choosing "own" options greatly reduces the number of positive samples for each target, so the optimization becomes hard. Because of the same reason the "word" target doesn't show good results, because the majority of the examples are negative. We also show that uniform splitting is a simple option that can be tried to outperform the phoneme baseline.

6. Conclusions

We investigated several classical and two new acoustic units for the keyword spotting task. It can be seen from our experiments, that the phoneme target is the best choice as a baseline, but it can be beaten in some cases. Also it's a useful trick to use or don't use the acoustic events similar to target ones, but which occur not in the target keyword. It especially helps when the number of positive examples is not very high.

7. References

1. J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden markov modeling for speaker-independent word spotting," in *International Conference on Acoustics, Speech, and Signal Processing*, May 1989, pp. 627–630 vol.1.

2. S. Myer and V. S. Tomar, "Efficient keyword spotting using time delay neural networks," in *Interspeech 2018*, 19*th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*., B. Yegnanarayana, Ed. ISCA, 2018, pp. 1264–1268. [Online]. Available: https://doi.org/10.21437/Interspeech.2018-1979

3. G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014. IEEE, 2014*, pp. 4087–4091. [Online]. Available: https://doi.org/10.1109/ICASSP.2014.6854370

4. M. Wu, S. Panchapagesan, M. Sun, J. Gu, R. Thomas, S. N. P. Vitaladevuni, B. Hoffmeister, and A. Mandal, "Monophone-based background modeling for two-stage on-device wake word detection" in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*. IEEE, 2018, pp. 5494–5498. [Online]. Available: https://doi.org/10.1109/ICASSP.2018.8462227

5. A. Kolesau and D. Šešok, "Voice activation systems for embedded devices: Systematic literature review," *Informatica*, vol. **31**, no. 1, pp. 65–88, 2020.

6. Wikipedia contributors, "Phoneme — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Phoneme&oldid=1006518035, 2021, [Online; accessed 21-February-2021].

7. Wikipedia contributors, "Syllable — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Syllable&oldid=1006686720, 2021, [Online; accessed 21-February-2021].

8. T. Zeppenfeld and A. H. Waibel, "A hybrid neural network, dynamic programming word spotter," in *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. **2**, March 1992, pp. 77–80.