IberSPEECH 2021
24-25 March 2021, Valladolid, Spain

Query-by-Example Spoken Term Detection using Attentive Pooling Networks at ALBAYZIN 2020 Evaluation: The AUDIAS-UAM System

Juan Ignacio Álvarez-Trejos[1], *Doroteo T. Toledano*[1]

[1]AUDIAS - Audio, Data Intelligence and Speech
Universidad Autónoma de Madrid

juani.alvarez@estudiante.uam.es, doroteo.torre@uam.es

Abstract

Query-by-example Spoken Term Detection (QbE-STD) is a key technology to harness the large amount of audiovisual content that is being stored and generated nowadays. Using audio example queries for STD has several advantages such as requiring less resources (both computational and linguistic) and resulting in less language-dependent systems. A further advantage is the possibility of developing neural end-to-end models. In this paper, we explore one of these models for QbE-STD. The model starts projecting the input pair formed by a query and a segment into fixed-length vector representations. Then, a distance between these vectors is calculated to generate a detection score. To learn similarities over the projected input pair, a two-way attention model, called attentive pooling networks, has been used. Both elements in the input pair can influence the vector representation of the other, paying more attention to the frames that contain key information of both the query and the occurrence. Our main objective is to explore if this model can find similarities regardless of the language used for training. We start showing the effectiveness of the proposed model on the Librispeech corpus, and then we evaluate it on the ALBAYZIN 2020 Search-on-Speech evaluation data.

Keywords: Query-by-example, Spoken term detection, End-to-end systems, Two-way attention, Attentive pooling networks

1. Introduction

Spoken term detection (STD) is defined as the task of retrieving audio segments which contain a query from an audio archive. If the query is in text form, i.e. keyword search [1], the task is typically solved based on automatic speech recognition (ASR) technology. If the query is an acoustic example (spoken query) we have Query-by-Example (QbE) STD. The QbE-STD task can be formulated as a detection task in which the input is an audio pair and the output is the list of hypothesis detected, which contains a detection score and the time intervals in which the detection resides. The most attractive feature of QbE-STD is that it is not necessary to transcribe the audios to text, and with this formulation of the problem, end-to-end architectures can be used, requiring less processing and taking better advantage of the amount of existing multimedia information.

Other widely used techniques such as Dynamic Time Warping (DTW) seek to directly compare the acoustic characteristics by constructing a frame-level similarity matrix [2]. The main idea of this method is to find the optimal warping path with the smallest distortion. DTW algorithm has been found to work best on high-level features, such as phone posteriorgrams [3]. However, these features are not available for low resource languages. Furthermore, the dynamic programming algorithm for similarity measure is time-consuming.

In this evaluation we apply a novel Two-Way attention mechanism also known as Attentive Pooling Networks. This method was successfully applied in the context of Query-by-example Spoken Term Detection in read speech in English and using word alignments [4]. Our objective is to analyze the performance of this approach under more realistic conditions, evaluating it on the scenario of natural speech in Spanish that is proposed in the Albayzin Search-on-Speech 2020 evaluation.

2. Attentive Pooling Networks for Query by Example

In this section, we present our primary system proposed for the Query-by-Example Spoken Term Detection (QbE-STD) task. This system is based on Attentive Pooling Networks, a recently proposed method for QbE-STD showing promising results [4].

2.1. System Description

From now on, we denote the spoken query, or more precisely the acoustic feature sequence, as $Q = \{q_1, q_2, ..., q_M\}$ and an audio segment where the query will be searched by $S = \{s_1, s_2, ..., s_N\}$, where M and N are the number of frames of the query and audio segment, respectively.

a. Embedding representation of audio segments

A Long Short-Term Memory (LSTM) network is used to obtain embedding representations of the two input audio segments. LSTMs are capable of storing temporal information, in theory, for long time spans [5]. Given a spoken query $Q = \{q_1, q_2, ..., q_M\}$, LSTM units project the query into a hidden state sequence $H_Q = \{h_1^Q, h_2^Q, ..., h_M^Q\}$, where h_M^Q contains information of the whole query. In the same way, the audio segment $S = \{s_1, s_2, ..., s_N\}$ is encoded into a second hidden state sequence $H_S = \{h_1^S, h_2^S, ..., h_N^S\}$. The same LSTM is used for input and query, so we expect that the representation of the query and the audio segment are in the same vector space.

b. Two-Way Attention: Attentive Pooling Networks

Now, we describe the Two-Way Attention mechanism, also called attentive pooling networks [4]. Figure 1 shows the structure of the system. First, the audio query $Q = \{q_1, q_2, ..., q_M\}$ and the audio segment $S = \{s_1, s_2, ..., s_N\}$ are encoded into the hidden vector sequences $H_Q = \{h_1^Q, h_2^Q, ..., h_M^Q\}$ and $H_S = \{h_1^S, h_2^S, ..., h_N^S\}$ by the shared RNNs to project the acoustic features into a vector space where they are more easily compared. Then the attention matrix G is computed as follows:

$$G = tanh(H_Q^T U H_S) \tag{1}$$

10.21437/IberSPEECH.2021-14

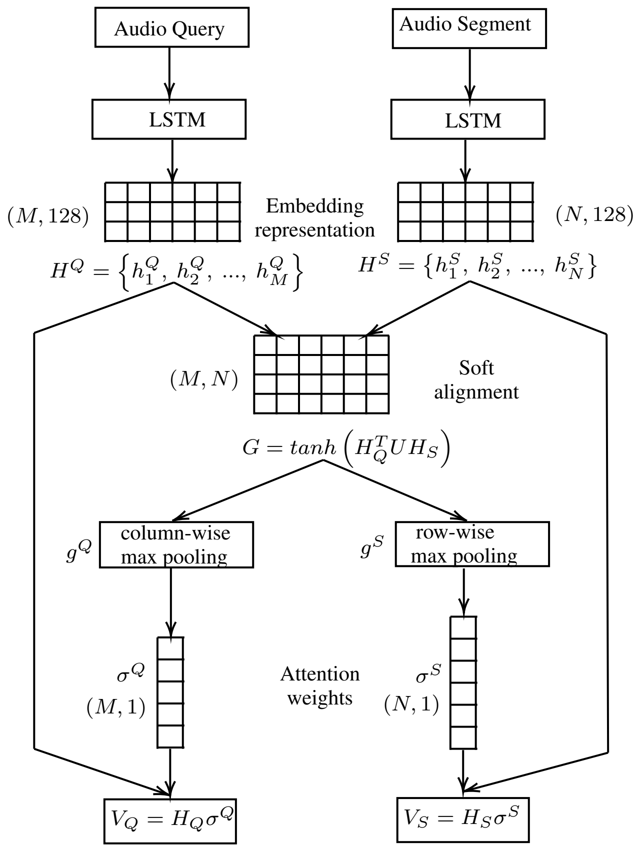

Figure 1: *Structure of Two-way attention (Attentive Pooling Networks) applied for QbE-STD in our Primary System*

This matrix is the main key to compare both, segment and query hidden state vector sequences. We can see matrix U as a measure of joint H_Q and H_S representations, and it is learned by training. To build a system more symmetric, we limit the measure matrix U to be a symmetric matrix, $U = U^T$. This allows to exchange the query and the segment without changing the result, because of the following relation.

$$H_Q^T U H_S = H_S^T U^T H_Q = H_S^T U H_Q \tag{2}$$

We initialize U with random normal samples with zero mean and 10^{-4} variance. Under these conditions, we can interpret G as a soft alignment score between each frame of the spoken query and the audio segment.

The next step is to apply a row-wise and column-wise max-pooling to the G matrix in order to generate the weight vectors $g^Q \in \mathbb{R}^M$ and $g^S \in \mathbb{R}^N$. This pooling is computed as follows.

$$[g^Q]_j = max_{1 \leq i \leq N}[G_{j,i}] \tag{3}$$

$$[g^S]_i = max_{1 \leq j \leq M}[G_{j,i}] \tag{4}$$

Note that the j-th element of vector g^Q represents the weight that we will apply to the j-th frame in the spoken query Q. Consequently, the result would be an estimation of which frames of the input sequences are actually important to make the comparison. Since that g^Q and g^S are attention vectors, it is necessary to normalize them with a softmax function to generate the final representation of attention vectors σ^Q and σ^S.

Finally, if we compute the dot product between the original hidden states and the attention vectors estimated, we have two representation vectors more comparable between them, because we have enhanced the parts that are really important to differentiate them. Vectors V_Q and V_S are computed as follows.

$$V_Q = H_Q\sigma^Q, V_S = H_S\sigma^S \tag{5}$$

c. Large-Margin Training

The whole system is trained using a large margin cost function (hinge loss). This is because the principal goal is to maximize the distance between classes and minimizing the intra class distance in each epoch. To achieve this, the training set is structured as groups formed by three elements: a spoken query $Q = \{q_1, q_2, ..., q_M\}$, a positive segment $S^{(P)} = \{s_1^{(P)}, s_2^{(P)}, ..., s_N^{(P)}\}$ and a negative segment $S^{(N)} = \{s_1^{(N)}, s_2^{(N)}, ..., s_N^{(N)}\}$. The positive segment contains the same word as the query, while negative segment is an aleatory audio segment taken from the training set that does not contain the same word as the query. Then, for each group we form the tuples $(Q, S^{(P)})$ and $(Q, S^{(N)})$ in order to calculate both of the attention matrices G for each tuple and the corresponding vector representations $(V_Q^{(P)}, V_S^{(P)})$ and $(V_Q^{(N)}, V_S^{(N)})$. Note that the V_Q vector representation is different depending on the audio segment for which it is computed, due to the Two-Way Attention. Cosine distance is used to measure how these tuples look alike. Cosine distance between V_Q an V_S is computed as follows.

$$l(V_Q, V_S) = (1 - cos(V_Q, V_S))/2 \tag{6}$$

We aim to minimize the distance between the query and the positive segment and to maximize distance between the query and the negative segment. For that a hinge objective function is used, defined as follows (M is the maximum possible distance, 1 in our case).

$$L_{hinge} = max\{0, M + l((V_Q^{(P)}, V_S^{(P)})) - l((V_Q^{(N)}, V_S^{(N)}))\} \tag{7}$$

3. Experimental Setup

As acoustic features we use 13 dimensional MFCC, which are extracted using the Kaldi toolkit [6] and Python Speech Features library with a sliding window with length of 0.025 seconds and a step of 0.01 seconds. These features are used in all the experiments proposed.

The shared RNNs consist of 2 layers with 128 LSTM units on all the models. Matrix U is a symmetric square matrix initialized with random normal values (with zero mean and a variance of 10^{-4}). As optimizer, we use Adam with a learning rate of 0.00005 and a minibatch of 128. We also use the WebRTC Voice Activity Detector (VAD) on all audio queries in order to remove all fragments of silence and random noise that could appear at the start and at the end of the audio recordings. All the neural networks are implemented with Pytorch and are trained for 4 epochs.

3.1. LibriSpeech Database description

a. LibriSpeech Training set

We use audio segments from the LibriSpeech Corpus [7]. We extract the segments from aligned utterances (with previously trained HMM models), so it is not necessary to use the VAD detector because each segment contains a complete word exactly. Furthermore, we choose those segments that have at least 6 phonemes and a duration between 0.5 and 1.0 seconds. Then, we randomly select 500 different words and we form a training sample taking 2 segments containing the same word (one for the query (Q) and other for the positive segment (S^P)) and 1 segment containing a different word for the negative segment (S^N). For each training epoch we have a total of 1000 of such groups, consisting in (Q, S^P, S^N).

b. LibriSpeech Test Set

We use a separate LibriSpeech test set divided into two subsets. First, we define a set with 150 words that appear in the training set as queries. We call these IV (In-Vocabulary) segments. Second, we have another set with 150 words, but in this case none of this words appear in the training set as queries. We call these OOV (Out-Of-Vocabulary) segments.

For each word in the full test set, we randomly select 20 different words to compare with (IV and OOV segments). We force to have at least 5 words equal to the query to ensure a sufficient number of positive comparisons. Like in the training set, we force each word to have at least 6 phonemes and a duration between 0.5 and 1.0 seconds. Finally, the LibriSpeech test set is formed by 3000 IV segments and 3000 OOV audio segments.

Mean Average Precision (MAP), the mean of the average precision in the range of recall for each query in the testing set and P@20, the precision considering the 20 best scores, are the evaluation metrics employed for LibriSpeech test Set.

3.2. Albayzin Search on Speech Database description

a. Development data

Training and development data provided by the evaluation organizers belong to the MAVIR and RTVE databases.

- For MAVIR database, development list of terms have about 375 different terms (375 OOV for our systems) whose length ranges from 5 to 27 single graphemes for Spoken Term Detection task. There is a total of 100 queries with three examples per query. Some of the queries have a fixed length of 3 seconds of audio, including a large amount of silence. We used webrtcVAD on all the segments to remove the silence before using them.
- For RTVE database, two different datasets are provided: *dev1* and *dev2*. Dataset *dev1* is not used in this paper. The dataset *dev2* consists of about 400 different terms (all are OOV for us) whose length ranges from 4 to 25 single graphemes. Like in MAVIR database, there is a total of 100 queries with three examples per query. We used webrtcVAD on the RTVE queries to reduce the large amount of silence appearing in some of them.

b. Test data

Three databases are provided by the organizers for system evaluation:

- MAVIR test speech data consists of about 200 different terms whose lengths range from 4 to 28 single graphemes. For the Query-by-Example Spoken Term Detection task, the systems implemented are tested with about 100 queries.
- RTVE test data consists of about 400 different terms of the RTVE program material given by the evaluation. RTVE test data has a total of about 400 different terms whose length ranges from 4 to 28 single graphemes. Again, this data set is composed by 100 queries and three examples per query.
- SPARL20 test data consists of a subset of Spanish parliament sessions. SPARL20 has a total of 200 different terms whose length ranges from 3 to 19 single graphemes. Like in the other testing sets, this set is conformed by 100 queries with three examples per query.

ATWV (Actual Term Weighted Value) and MTWV (Maximum Term Weighted Value) are the metrics proposed by the evaluation. We use the VAD detector webRTCVAD on all the test data queries.

4. Results on LibriSpeech test set

In this section we describe two different experiments performed and their corresponding results when evaluated on the LibriSpeech test Set. Both experiments differ in the alignment of the segment used to search for the query.

In experiment 1, we train the Two-Way attention model with LibriSpeech Training set without introducing noise and using the word alignments to train and evaluate the system. This evaluation is highly unrealistic since it requires to have the word alignments of the segments on which the query is searched. For that reason we consider an alternative setup.

In experiment 2, we aim to evaluate the performance of the system trying to simulate the more realistic setup of not having the word alignments of the segments on which the query is searched. We start from the same set used in experiment 1 and, for each pair (Q, S) we look for two random words from the entire subset not coincident with the words contained in Q and S. Then, we concatenate these words to the beginning and end of the segment, and trim the segment to a fixed length of N = 150 frames. In this way the segment does not contain a perfectly aligned word. This technique has been applied both for training and testing, so that this experiment simulates the possibility of not having word alingments in both training and test.

Table 1: *Performance of QBE-STD on LibriSpeech Testing Set*

Exp.	MAP (IVs)	MAP (OOVs)	MAP (Total)	P@20
Exp.1	0.981	0.971	**0.976**	**0.23**
Exp.2	0.972	0.953	0.962	0.06

As can be seen in the Table 1, better results are always obtained in experiment 1. As expected, results worsens when the word alignments are not available. However, the results obtained without word alignments (exp. 2) are still very good, since the total MAP is very close to one, indicating that the system is committing very few false positives. When considering the precision for the 20-best scores, the degradation in performance becomes more clear. Our results are much better than those reported in the original paper proposing this approach [4], most probably because testing was not performed in the same conditions. In particular, we have forced 5 of the 20 segments compared against each query to contain the same word as the query, while this was not enforced in the original paper.

5. Albayzin Search-on-Speech 2020 QbE-STD systems and results

This section describes the systems submitted to the 2020 edition of ALBAYZIN Search-on-Speech Query-by-Example Spoken Term Detection (QbE-STD) evaluation and the results obtained with them.

5.1. System Description

All of the systems submitted are based on the Two-Way attention mechanism (Attentive Pooling Networks) and trained on LibriSpeech (an English database). In some of the systems development data in Spanish has been used to try to adapt the system to the language of the evaluation. For each system, threshold was set to make ATWV as close as possible to MTWV on development data.

a. Primary System

For the primary system we have used the neural networks trained in LibriSpeech experiment 1, since we have not found improvements on development data by adding spoken word segments to the beginning and end of the segment to be compared (experiment 2). For this primary system, we have retrained the neural networks for two epochs with the MAVIR development set. The goal was to apply transfer learning [8] to adapt the system trained on a large database in English with a small database in Spanish.

As explained in the experimental setup, we first apply the WebRTC Voice Activity Detector to the 100 queries and then we go through the entire segments to be analyzed with an adaptive sliding window with a length dependent on the length of the query, since we expect the ocurrences of the query to be of similar size to the query.

b. Contrastive System 1

In this case, we have used the neural networks trained from LibriSpeech experiment 1 again, this time without modifying the weights with data in Spanish. We have also used an adaptive window length as in the primary system.

c. Contrastive System 2

Contrastive system 2 is almost identical to contrastive system 1, they only differ in that in this case we are applying a z-score normalization approach using

$$z = \frac{x - \mu}{\sigma} \tag{8}$$

Where μ is the mean of the scores obtained and σ is the standard deviation of these. We wanted to evaluate if there was a score normalization issue with this contrastive system.

5.2. Results on Albayzin Search-on-Speech 2020 data

Table 2 presents development and test results obtained on the evaluation data. The first conclusion is that we have not been able to successfully apply the Two-Way attention mechanism (or Attentive Pooling networks) on the evaluation data. All results present very low and even negatives ATWVs. Even without taking into account threshold setting (looking at MTWV) results are poor. There seem to be a number of unresolved issues in the transition from a read speech scenario in English (LibriSpeech) with word-aligned data to a natural speech scenario in Spanish (MAVIR, RTVE and SPARL20) without word-aligned data that have produced these results.

As can be seen in the table 2, better results have generally been obtained with the primary system, which indicates that there is some improvement when retraining for a few epochs with audios in Spanish.

In the two contrastive systems, there is an obvious problem in setting the detection threshold, as the MTWV and ATWV values differ greatly. Score normalization did not provide consistent improvements either, as can be seen in the results for the contrastive 2 system.

Table 2: *Performance of QBE-STD Primary System (PRI), Contrastive System 1 (CON1) and Contrastive System 2 (CON2) on Albayzin2020 development and test data*

Dataset	**System**	**MTWV**	**ATWV**
MAVIR DEV	PRI	**0.0533**	**0.0491**
	CON1	0.0160	-38.5775
	CON2	0.0000	-158.2873
MAVIR TEST	PRI	**0.0126**	**-0.1061**
	CON1	0.0000	-393.5610
	CON2	0.0000	-38.5959
RTVE DEV	PRI	**0.0465**	**0.0465**
	CON1	0.0414	-76.0473
	CON2	0.0000	-51.9993
RTVE TEST	PRI	**0.0209**	-115.7086
	CON1	**0.0209**	-88.3716
	CON2	0.0000	**-16.5831**
SPARL20 TEST	PRI	0.0107	**0.0107**
	CON1	**0.0306**	-34.2099
	CON2	0.0000	-103.6805

6. Conclusions

In this evaluation we have tried to apply a novel Two-Way attention mechanism also known as Attentive Pooling Networks. This approach was successfully applied in the context of Query-by-Example Spoken Term Detection in read speech in English (LibriSpeech) and using word alignments [4]. We have been able to reproduce good results on this scenario and have tried to improve the system by simulating the more realistic scenario of not having word alignments. However, this approach has not helped when transitioning from this scenario to the scenario of natural speech in Spanish that is proposed in the Albayzin Search-on-Speech 2020 evaluation. We have obtained limited improvements when retraining the system on a small amount of Spanish data but, in the end, the results obtained on the evaluation data are poor compared to other more classic alternatives. Anyway, we still consider that this approach has the potential to compete in results with these more classical approaches, and we will continue exploring it in the future.

7. Acknowledgements

Work developed under project DSForSec (RTI2018-098091-BI00), funded by the Ministry of Science, Innovation and Universities of Spain and FEDER.

8. References

[1] D. R. Miller, M. Kleber, C. L. Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," *International Speech Communication Association - 8th Annual Conference of the International Speech Communication Association, Interspeech 2007*, vol. 3, pp. 1965–1968, 2007.

[2] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[3] T. J. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," *Proceedings of the 2009 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2009*, pp. 421–426, 2009.

[4] K. Zhang, Z. Wu, J. Jia, H. Meng, and B. Song, "Query-by-example spoken term detection using attentive pooling networks," *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2019*, pp. 1267–1272, 2019.

[5] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, 2011.

[7] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2015-August, pp. 5206–5210, 2015.

[8] S. Panigrahi, A. Nanda, and T. Swarnkar, "A Survey on Transfer Learning," *Smart Innovation, Systems and Technologies*, vol. 194, no. 10, pp. 781–789, 2010.