Weakly Supervised One-Shot Detection with Attention Siamese Networks

Gil Keren[1,2]**, Maximilian Schmitt**[1]**, Thomas Kehrenberg**[1]**, Björn Schuller**[1,3]

[1] ZD.B Chair of Embedded Intelligence for Health Care and Wellbeing,
University of Augsburg, Germany
[2] Chair of Complex and Intelligent Systems, University of Passau, Germany
[3] GLAM – Group on Language, Audio & Music, Imperial College London, UK
cruvadom@gmail.com

Abstract

We consider the task of weakly supervised one-shot detection. In this task, we attempt to perform a detection task over a set of unseen classes, when training only using weak binary labels that indicate the existence of a class instance in a given example. The model is conditioned on a single exemplar of an unseen class and a target example that may or may not contain an instance of the same class as the exemplar. A similarity map is computed by using a Siamese neural network to map the exemplar and regions of the target example to a latent representation space and then computing cosine similarity scores between representations. An attention mechanism weights different regions in the target example, and enables learning of the one-shot detection task using the weaker labels alone. The model can be applied to detection tasks from different domains, including computer vision object detection. We evaluate our attention Siamese networks on a one-shot detection task from the audio domain, where it detects audio keywords in spoken utterances. Our model considerably outperforms a baseline approach and yields a 42.6% average precision for detection across 10 unseen classes. Moreover, architectural developments from computer vision object detection models such as a region proposal network can be incorporated into the model architecture, and results show that performance is expected to improve by doing so.

1 Introduction

Detection models proposed in the last few years have managed to considerably improve the state-of-the-art performance and execution speed for object detection tasks. However, the success of these proposed models is due to, among other factors, the use of large-scale labelled datasets that contain a large number of labelled examples for a limited number of classes (Deng et al., 2009; Lin et al., 2014). Humans, however, are required in daily life to correctly identify and localise a much larger number of classes.

Models that perform one-shot learning attempt to overcome the dependency on large amounts of class-specific labelled data by generalising from a single exemplar to other members of its object class. Recent works have shown that good performance of one-shot or few-shot learning models can be achieved by matching the train conditions to better reflect the requirements of a one-shot or few-shot learning model at evaluation time (Koch et al., 2015; Vinyals et al., 2016). Specifically, at every training iteration, the model is given a small number of examples from a small number of classes, and emits predictions to other examples of these classes. Training in such manner, using a large enough number of classes, was observed to yield generalisation at evaluation time to unseen classes. However, applying this approach for a one-shot localisation task may require a training

corpus that contains bounding box information for a very large number of object classes. As bounding box labels are in general harder or more expensive to get, corpora with bounding box information are often smaller than ones that only contain weaker labels.

Instead of relying on a large enough corpus that contains bounding box information, we attempt to perform one-shot detection using weaker labels alone. For the same reasons, previous work was done on learning object detection using image-level labels alone (Bilen and Vedaldi, 2016; Teh et al., 2016). In the absence of bounding boxes at training time, context-based attention models might be good candidates for learning localisation information in a weakly supervised manner, as these models previously demonstrated the ability to focus on relevant parts of the input element. Examples from different modalities include attending to the image part corresponding to a given word (Xu et al., 2015), the audio segment corresponding to a given text part (Chorowski et al., 2015) and the part in a source sentence corresponding to a word in a translated sentence (Bahdanau et al., 2015). These models are typically not trained with any localisation information. Instead, the localisation behaviours are learnt in a weakly supervised manner from weaker labels, i.e., labels that do not contain localisation information.

In this work, we present a one-shot detection model that does not use any localisation labels, e.g., bounding boxes. Instead of using bounding boxes as a supervision signal, we only use binary labels that indicate the existence of a certain object in the example of interest, and learn the localisation information using an attention mechanism in the manner described below. Our model takes a single example of a given class, which we name the *exemplar*, and a *target example* that may or may not contain an instance of same class as the exemplar. We adapt the Siamese neural networks framework (Bromley et al., 1993; Koch et al., 2015) and apply it in a convolutional manner, to compute the similarity between the exemplar and every location of the target example. The similarity scores of the different parts of the target example are normalised to be the attention weights, and the latter are used in a weighted sum that produces a single score for the existence of the exemplar in the target example.

We evaluate the proposed model on the task of spoken term detection (Hazen et al., 2009; Chen et al., 2015; Parada et al., 2009), which is a detection task from the audio domain. Specifically, we use audio utterances as the target examples and short audio recordings of different keywords as the exemplars. The task is then, for each utterance, to detect the keywords that appear in this utterance, including a bounding box (an interval) indicating the position of the keyword in the utterance. We note that the same model can be applied in a similar way for computer vision object detection, where exemplars are small images of objects (i.e., the image that is inside the bounding box) and the target examples are larger images. The model's performance is evaluated using an average precision metric, similar to standard metrics used in object detection in computer vision (Everingham et al., 2010; Lin et al., 2014). Results show that our one-shot detection model considerably outperforms a dynamic time warping (Joder et al., 2012) baseline, and identifies and localises unseen classes (keywords) with good performance. Moreover, architectural developments from computer vision object detection models can be incorporated into the model's architecture, and results show that performance is expected to improve by doing so. We conclude that the proposed model is capable of performing one-shot detection tasks, without using any localisation labels at training time, but rather learn the localisation information in a weakly supervised manner. Future work should further develop attention Siamese networks and apply them to computer vision object detection, while incorporating recent architectural developments.

The rest of the paper is organised as follows. Section 2 describes related work about one-shot and weakly supervised learning. Our attention Siamese networks model is described in Section 3. Experiments and evaluations are conducted in Section 4, while we draw final conclusions in Section 5.

2 Related Work

Related one-shot learning models were presented in recent years, for various tasks. In Koch et al. (2015), the task of image verification was considered. A Siamese network processes two images and outputs a score predicting whether these images are of the same class. In Vinyals et al. (2016), A model for one shot classification is presented. This work focuses on the idea of matching training conditions to the conditions dictated for one-shot learning at test time. Specifically, at training time, the model is presented with a small dataset from a small number of classes. Then, an attention
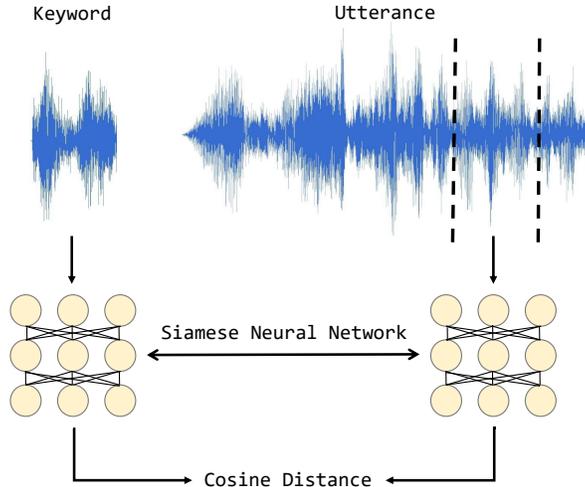
Figure 1: The computation of a similarity score between an exemplar and a target location, for the case of detecting audio keywords in longer utterances. Both the exemplar (keyword) and the target location (part of the utterance) are mapped into a representation space using a Siamese neural network, and a cosine distance between the representations is calculated.

mechanism conditioned on a target image attends to specific members of the presented dataset, and decides accordingly on the predicted class for the target example. In another model for a few-shot recognition task (Hariharan and Girshick, 2016), a learner tunes its feature representation on a set of base classes that have many training instances, to better perform on classes with a small number of examples. For the task of one-shot detection, A computer vision model for few-shot object detection was presented in Dong et al. (2017). This model is comprised of a pipeline that includes finding additional training examples in a large unlabelled dataset.

Weakly supervised object detection was considered in previous work. Wang et al. (2014) use a pre-trained convolutional neural network (CNN) to describe image regions and then learn object categories as corresponding visual topics. Most similar to our approach, the model in Teh et al. (2016) computes an attention score for every location in an image. These scores are then combined to one feature vector describing an image, that is used for classifying the image. Localisation is done using the attention weights. Similarly, the work presented in Bilen and Vedaldi (2016) starts from a CNN pre-trained for image classification on a large dataset, then computes scores for each class at each location. These scores are combined into a single image level score, and the network is optimised for the classification task. Detection is again performed according to the class location scores. However, all of the above models depend on a predefined set of of classes, and are not suited for one-shot detection. In the audio domain, a related model for query-by-example spoken term detection was proposed concurrent to our work (Ao and Lee, 2017). This model uses an long short-term memory (LSTM) network for scoring the existence of a keyword in different utterance locations. These scores are combined for a single score for the utterance, that is used for training on a binary classification task.

3 Method

Our model takes an *exemplar* x – a single instance of a class of interest and a *target example* B, which may or may not contain an instance of the same class as the exemplar. The target example is normally spatially larger than the exemplar, so that the exemplar could be compared to different locations in the target example. We name different locations in the target image *target locations*. The output of the model is a *similarity map* $s(x, B)$ over target locations, such that $s_l(x, B)$ is a measure of the similarity of location l in the target example to the exemplar. For simplicity of notation, we write s_l instead of $s_l(x, B)$ when the context is clear.

3

3.1 Similarity Scores

To obtain a meaningful similarity measure between the exemplar and target locations, we do not require the raw representations to be within a small distance from one another. Instead, we would like to map both exemplar and target locations to a latent representation space, and optimise for similarity or dissimilarity in the representation space. Therefore, the computation of s is modelled using the cosine distance between outputs of a Siamese neural network (Bromley et al., 1993; Koch et al., 2015):

$$s_l = \frac{f_\theta(x) \cdot f_\theta(B_l)}{\|f_\theta(x)\|\|f_\theta(B_l)\|}, \tag{1}$$

where B_l is location l in the target example B, and f_θ is a neural network parameterised by θ, that is embedding both the exemplar and target locations into the latent embedded space. The similarity score s_l is in the unit interval. See Figure 1 for a visualisation of the similarity score computation for one target location. The above computation can be seen as (and was implemented as) a convolutional application of the Siamese network (Bertinetto et al., 2016) across all locations of the target example, and application of the cosine distance to the resulting map. In case f_θ is a CNN, computation can be reduced by applying f_θ once on the whole target example and extracting the relevant parts of the feature maps that correspond to each location, as was done in Girshick (2015).

3.2 Weakly Supervised Detection

We denote every *Siamese pair* (x, B_l) as either positive, if the target location B_l contains an instance of the same class as the exemplar, or negative, otherwise. Similarly, we denote every *exemplar-target* pair (x, B) as either positive, if an instance of the same class as x appears in some location in B, or negative, otherwise. The goal of training is to increase similarity scores $s_l(x, B_l)$ for positive Siamese pairs and decrease it for negative Siamese pairs. Recall, that we do not have access to labels containing any localisation information, but rather only binary labels $y_{x,B} \in \{0, 1\}$ that indicate whether the exemplar-target pair (x, B) is positive or negative.

In the absence of localisation labels, an additional credit assignment problem arises during training – namely, which Siamese pairs should be assigned a greater similarity score and which should be assigned a smaller similarity score. For negative (x, B) pairs, the answer is simple – similarity score s_l should be reduced to zero for all locations l (all Siamese pairs are negative). However, for positive (x, B) pairs, the similarity score should be increased for locations that contain the instance of the same class as the exemplar (positive Siamese pairs), and decreased for locations that do not contain such instance (negative Siamese pairs), without any labels that contain location specific information.

We attempt to overcome this issue by making locations in a positive exemplar-target pair compete for a high similarity score, as explained below. Related methods for weakly supervised detection were introduced in Bilen and Vedaldi (2016) and Teh et al. (2016), but these are not suited for one-shot learning (see Section 3.3). We compute attention weights by applying softmax normalisation to the similarity map

$$w_l = \frac{\exp(s_l/T)}{\sum_{l'} \exp(s_{l'}/T)}, \tag{2}$$

where T is the softmax temperature. The attention weights are then used to compute a single similarity score for an exemplar-target pair

$$\hat{y}_{x,B} = \sum_l w_l s_l, \tag{3}$$

which is again in the unit interval, and the loss for a single pair (x, B) is

$$\ell(x, B) = (\hat{y}_{x,B} - y_{x,B})^2.$$

Using the gradient of the softmax function, we can compute

$$\frac{\partial \hat{y}}{\partial s_l} = w_l + \frac{s_l w_l(1 - w_l)}{T} - \sum_{l' \neq l} \frac{s_{l'} w_l w_{l'}}{T} = w_l(1 + \frac{s_l}{T} - \frac{1}{T}\sum_{l'} w_l s_l) = w_l(1 + \frac{s_l - \hat{y}_{x,B}}{T}),$$

and the gradient of the above loss function with respect to similarity scores is

$$\frac{\partial \ell}{\partial s_l}(x, B) = 2(\hat{y}_{x,B} - y_{x,B})w_l(1 + \frac{s_l - \hat{y}_{x,B}}{T}). \tag{4}$$

For positive exemplar-target pairs ($y_{x,B} = 1$), the gradient in Eq. 4 is monotonic decreasing in s_l and w_l. Therefore, as the step size is proportional to the negative gradient, propagating this gradient back to network parameters during training will cause similarity scores for Siamese pairs that already have higher similarity scores to increase even more, compared to similarity scores of Siamese pairs with already lower similarity scores. This is a self-reinforcing loop: in positive exemplar-target pairs, Siamese pairs with high similarity scores will get even higher similarity scores during training, therefore also higher attention weights and again a more negative gradient that will increase similarity scores for these locations even more. Similarly, similarity scores for Siamese pairs with already lower similarity score will be increased less compared to similarity scores of other Siamese pairs, causing lower attention weights and again a less negative gradient. See Figure 2 for a visualisation of the relation between the similarity scores and the gradient propagated to them.

According to the above, it is enough to only make positive Siamese pairs have slightly larger similarity scores than negative Siamese pairs, and the self-reinforcing loop will increase this difference. The intuition here is that positive Siamese pairs have in general more in common with each other than negative pairs have in common with each other, in the sense that gradient updates based on minibatches will increase similarity scores of positive Siamese pairs more than negative ones. This difference between positive and negative Siamese pairs should be enough for the self-reinforcing loop to begin and eventually assign considerably larger similarity scores to positive Siamese pairs compared to negative ones.

3.3 One-Shot Learning

By conditioning the model only on the exemplar and the target example and not on any explicit class label, we allow the model to better generalise to unseen classes. Indeed, exemplars from unseen classes can share characteristics with already seen exemplars, aiding the model to make correct predictions for examples of unseen classes. This can be seen as the model implicitly learning to represent a class by its examples, then being able to generalise to similar classes in this class representation space. Related approaches were used in Koch et al. (2015) and Vinyals et al. (2016) for the tasks of one-shot object verification and one-shot classification.

We define T to be the uniform distribution of classes available at training and evaluation time. We denote with X_L the distribution of exemplar-target pairs (x, B) such that x belongs to class L, and B contains an instance of class L with a probability of 0.5. At training time, we sample a class from the distribution T' that is uniform over the classes available at training time, and try to minimize the loss over exemplar-target pairs sampled from X_L that corresponds to the appropriate class. Our goal of training is then choosing model parameters Θ such that

$$\Theta = \arg\min_\theta E_{L \sim T'}[E_{(x,B) \sim X_L} \ell(x, B)]]. \tag{5}$$

Training the model with Eq. 5 should yield a model which works well when sampling $L \sim T$, instead of $L \sim T'$, such that classes that were not seen in training time are included. For demonstrating one-shot learning abilities, we evaluate our model in the experiments section only on classes that were not included on the training set. For predicting well on unseen classes, our model does not need any fine-tuning that adjusts the model to the new classes.

3.4 Detection

For a detection task, given a target example and a set of possible classes, we are interested in answering the question which of the possible classes appear in the target example, and what location they appear in. Our model takes as input a target example and an exemplar of one particular class, therefore in order to consider all possible classes we need to feed our trained model with the target example, together with an exemplar of each possible class. This can be done in a more computationally efficient way though, where $f_\theta(B_l)$ from Eq. 1 is only computed once for each target location, and compared with the embeddings of all the different exemplars.

To decide upon detections, a threshold t is used. For every exemplar-target pair (x, B), we say that B contains an instance of the same class as x if $\hat{y}_{x,B} > t$. For such exemplar-target pairs, we say that the location of the detection corresponds to the target location l with the highest similarity score $s_l(x, B_l)$. Note that here we make a simplifying assumption that an instance of the class of the exemplar may appear in the target example at most once. This can be generalised to cases where this
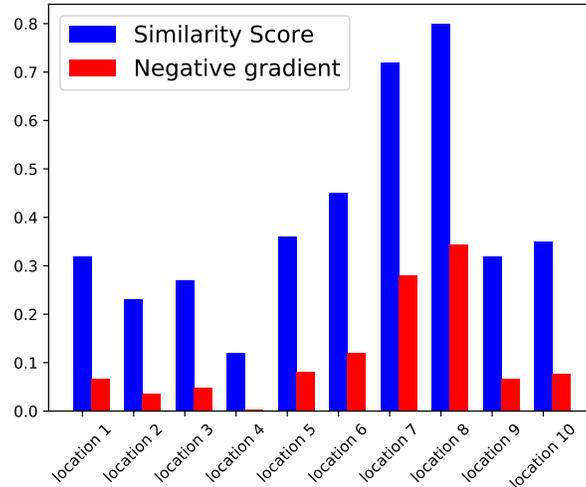
Figure 2: The reason for the self-reinforcing loop: example relation between similarity scores s_l and the negative gradients $-\frac{\partial \ell}{\partial s_l}$, in a positive exemplar-target pair with ten different locations, computed according to Eq. 4 for an example vector of similarity scores. The negative gradient is increasing with the similarity scores. As a result, locations with high similarity scores will get even higher similarity scores during training, compared to other locations.

assumption is not true by emitting multiple detections if a high similarity score is present in multiple locations, but for simplicity we stick to the case where the assumption holds. See Section 4.4 for more details about the process of emitting detections and their evaluation.

4   Experiments

We evaluate our weakly supervised detection model on the task of detecting audio keywords in longer audio utterances, also known as query-by-example spoken term detection (Hazen et al., 2009; Parada et al., 2009; Wöllmer et al., 2009b, 2010; Chen et al., 2015). This task can be seen as the equivalent of computer vision object detection in the speech domain. In this task, every word represents a class. Each audio keyword is an exemplar of some class, and the longer audio utterances are the target examples. The goal is to determine whether the same word as the keyword appears in the utterance, and determine the bounding box location of the appearance, in seconds, in case the word appears in the utterance.

4.1   Data

We construct a large-scale dataset for our keyword detection task, using two separate existing corpora: a speech recognition corpus and an audio keywords corpus. The audio keywords were downloaded from the Shtooka project website `http://www.shtooka.net`. All keywords are in English, and are less than one second long. Every audio keyword is unique, in the sense that no word appears in two different audio recordings. Each keyword was allocated to either the training, the validation or the test set (the split allocation can be found in `http://www.openaudio.eu/`). By doing so, we make sure we evaluate the model on detection of keywords that were not seen during the training phase, which is a one-shot learning task. The textual from of words can appear as a part of other, longer words (for example, 'the' is a part of 'their' and 'further'), which results in an undefined desired behaviour for the model. For this reason, we chose not to use short words, that are more prone to this issue, and we only use words that consist of four letters or more.

All audio utterances we use are from the Librispeech corpus (Panayotov et al., 2015). The Librispeech corpus contains 1000 hours of annotated English speech, from 2484 speakers. Each utterance in the Librispeech corpus was allocated to one of our training, validation or test sets, according to the

official split of this corpus, which is gender balanced. We cut each utterance to be exactly five seconds long.

Our training dataset is comprised of keyword-utterance pairs, with binary labels that indicate the existence of the keyword in the utterance. We use the term *positive pairs* to refer to keyword-utterance pairs where the keyword appears in the utterance (according to the utterances' transcriptions), and *negative pairs* to refer to other pairs. Note that the keyword and the utterance are always from different recordings, made by different speakers. For constructing our training set, we use all positive keyword-utterance pairs where both the keyword and utterance were allocated to the training set. In order to balance the two classes in the training set, we make sure every keyword appears in the same number of positive and negative pairs. We do this by randomly sampling a number of training utterances that do not contain the keyword and add the resulting negative pairs to the training set. In total, our training set contains 330,018 keyword-utterance pairs.

For constructing the validation and test sets, we first add all positive keyword-utterance pairs that belong to the appropriate evaluation set (validation or test). For evaluating the model on detection over a number of unseen possible classes (different unseen classes every time though), for every positive keyword-utterance pair in the evaluation set we add another n negative keyword-utterance pairs that comprise of the same utterance as in the positive pair. The keywords and utterances in the negative pairs we add are also from the same set (validation or test) as the ones in the positive pair. Note that the more negative keyword-utterance pairs we add to the model, the more false positives we are likely to find, that should impair the overall performance of the detection model. We create the validation and test set as described above with $n \in \{10, 20, 50\}$, and we name the resulting test sets $test10$, $test20$ and $test50$ respectively. See Section 4.4 for details about the detection task evaluation method.

Labels regarding the temporal location of a keywords in an utterance were extracted using forced alignment. Specifically, we used the Montreal Forced Aligner (McAuliffe et al., 2017) with default parameters. These labels were used for creating ground truth bounding boxes, that are used when evaluating the detection model on the validation and test sets. However, these labels were not used at training time, as in this work we consider a weakly supervised model.

As some words are more common than others, every keyword has a different number of keyword-utterance pairs that it is a part of, which can result in a small number of keywords dominate the training or evaluation procedure. To counter this effect, in each of the training, validation and test sets, we count the number of keyword-utterance pairs that each keyword appears in, and we use only keywords that are below the 85th percentile in this count. Overall, our training, validation and test sets contain 3592, 259 and 285 audio keywords respectively.

4.2 Input Features and Network Specifications

We represent the audio recordings as their spectrograms. For each 16 kHz waveform that corresponds to a keyword or utterance, we apply a short-time Fourier transform (STFT) over frames of 25 ms, shifted by 10 ms. We only use the magnitude information from the STFT results, and ignore the phase information as normally done in machine learning applications for audio analysis. Overall, this process results in extracting 201 features for every frame.

Our model is based on computing similarity scores between an exemplar and locations in a target example. The representations of the exemplar and the location in the target example are computed using a neural network f_θ, as appears in Eq. 1. The neural network f_θ must match the characteristics of the data used. In our experiments, we use this model for the detection of audio keywords in longer audio utterances. Therefore, we chose the neural network f_θ to be a CNN that is comprised of one-dimensional convolutions. The model can be used in a similar way for computer vision object detection, with different specifications for f_θ, such as two dimensional convolutions, etc. In our case of audio keywords detection, the network f_θ is a fairly standard CNN, similar to the networks used in Keren et al. (2017b) and Keren et al. (2017a) and has the following specifications. The network is comprised of eight convolutional layers. Each convolutional layer is performing a one-dimensional convolution using a kernel size of five frames, with a stride of one frame. The first four convolutional layers are comprised of 256 feature maps, while the last four convolutional layers are comprised of 512 feature maps. To shorten the temporal length of the sequences, after every second convolutional layer we apply a max-pooling operation over non-overlapping groups

Model	Set	AP[%]	Pr'@0.5[%]	Pr'@0.9[%]	Pr'@0.99[%]
Attention Siamese Networks	$test10$	42.6	73.1	25.9	12.7
	$test20$	38.3	50.9	16.0	6.5
	$test50$	23.6	29.9	5.8	2.6
Dynamic Time Warping	$test10$	8.9	14.6	11.1	10.4
	$test20$	6.0	7.8	5.8	5.5
	$test50$	3.7	3.3	2.4	2.3

Table 1: Evaluation of attention Siamese networks and the dynamic time warping model, for the detection task and the binary classification task over 10, 20 and 50 unseen classes. Average precision (AP) and precision at given recall levels (Pr'@x).

of two frames. Every convolutional layer is followed by a batch normalisation operation (Ioffe and Szegedy, 2015) and rectified-linear activation function. The output of the last max-pooling operation is flattened into a one-dimensional representation for the keyword or location of the target example. For a keyword of 0.8 seconds, this results in a 2048-dimensional representation (though the length of this representation varies with the length of the keyword). When computing the attention weights according to Eq. 2, a softmax temperature of $T = \frac{1}{3}$ is used. The network is trained according to Eq. 5 using Stochastic Gradient Descent with a learning rate of 0.1 and minibatch size of 64 keyword-utterance pairs. Training is stopped when performance is best on the validation set, which was also used for deciding upon hyperparameters.

4.3 Dynamic Time Warping

We use dynamic time warping (DTW) as a baseline to compare our proposed method to. DTW matches sequences of different lengths with each other (Wöllmer et al., 2009a). As the duration of the articulation of a word usually differs between speakers and situations, DTW is a well-established approach to query-by-example spoken term detection (Joder et al., 2012). As it does not require any training phase, it is suitable for our one-shot query-by-example paradigm. As previously done for spoken term detection with DTW (Joder et al., 2012), we use Mel-frequency cepstral coefficients (MFCCs) as acoustic features. We use the coefficients one to twelve, extracted from frames of 20 ms shifted by 10 ms with the toolkit OPENSMILE (Eyben et al., 2013). The DTW algorithm finds the shortest path between the MFCC representation over time of each keyword and a given segment from an utterance. DTW then returns a cost value describing the sum of the Euclidean distances of the shortest path between the sequences. This cost function C is converted to the probability of a match, which we use as the similarity measure between the keyword and the location in the utterance

$$s_l = \exp \frac{-C}{\sigma},$$ (6)

with the parameter σ, where 50 was found to yield best results on the validation set. The confidence was computed for each possible location within the utterance, i.e., for each 10 ms step as this is the hopsize of the MFCC feature vectors.

4.4 Evaluation

Both our proposed attention Siamese networks and the DTW algorithms output a similarity score s_l between the keyword x and every location l in a utterance B. For our Siamese attention network model, we emit a detection with confidence $\hat{y}_{x,B}$ for a keyword-utterance pair (x, B) whenever $\hat{y}_{x,B} > t$, where $\hat{y}_{x,B}$ is computed as in Eq. 3, and t is a threshold that is determined using the validation set. For the DTW baseline, there exists no such score for a keyword-utterance pair. Therefore, for this model we define $\hat{y}_{x,B} = \max_l s_l(x, B)$, and again emit a detection with confidence $\hat{y}_{x,B}$ whenever $\hat{y}_{x,B} > t$, where t is again chosen using the validation set. For both models, we denote the location with the maximal similarity

$$l^{max} = \arg \max_l s_l,$$

and we denote l^{max}_{start} and l^{max}_{end} as the start and the end of the location in the utterance that corresponds to the location l^{max} in seconds from the beginning of the utterance. The final bounding box for the
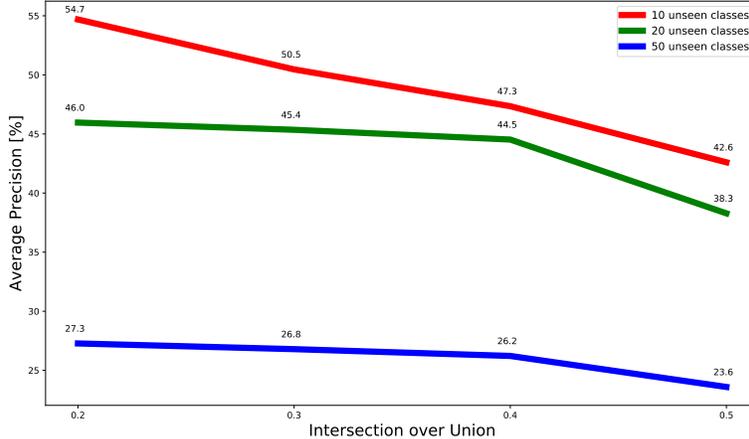
Figure 3: Average precision (AP) of the attention Siamese Networks for the audio keywords detection task over 10, 20 and 50 unseen classes with different intersection over union (IoU) thresholds.

detection is $\left(l_{start}^{max} + a, l_{end}^{max} + b\right)$, where a and b are shift parameters that are again determined on the validation set for each model separately.

Given a model's detections and the ground truth bounding boxes, we compute the model's performance in a manner similar to the performance of object detection models in computer vision, using an intersection over union (IoU) threshold of 0.5 (Everingham et al., 2010). The only difference between the performance measure in Everingham et al. (2010) and our performance measure is that in our case we, compute the average precision (AP) for detection of all classes (all words) together, instead of averaging the AP computed for each class separately (mAP). The reason for this deviation from the well established performance measure is that for some of the classes we use in evaluation time, we only have a small number of ground truth bounding boxes available (some keywords appear in less than ten utterances). Indeed, when the number of ground truth boxes is small, the computation of precision values over fine grained recall values is less meaningful.

In addition to detection, both our proposed model and the DTW baseline perform binary classification of exemplar-utterance pairs, to determine whether the keyword appears in the utterance. We evaluate the performance for this task by computing the precision at recall levels of 0.5, 0.9 and 0.99, where the $\hat{y}_{x,B}$ is the score assigned to each keyword-utterance pair.

DTW is a natural algorithm to compare our model to, due to similarities between the two approaches. Indeed, both models output similarity scores for different locations in the utterance, and do not use any localisation labels. We did not find any other existing one-shot detection systems that do not use any localisation labels and are suitable for the task of spoken term detection, therefore comparison is made only between our proposed method and DTW. Further quantitative evaluation of our model is done below, in addition to the comparison to DTW.

Results of all performance measures for the attention Siamese networks and the DTW model appear in Table 1. Each model was evaluated on the three test sets, $test10$, $test20$ and $test50$ that were constructed in Section 4.1 and correspond to a detection problem over 10, 20 or 50 unseen classes respectively. As expected, for both models, performance on the $test10$ is better than performance on the $test20$ set, which is better than the one on the $test50$ set. Indeed, as the ratio between the number of negative and positive keyword-utterance pairs increases, each model produces more false positives with respect to the threshold t, which impairs the performance for the detection task as well as for the binary classification task. Furthermore, our proposed Siamese attention networks outperform the DTW baseline by a large margin in all performance measures, for the three different test sets. Specifically, our proposed model yielded AP scores of 42.6%, 38.3% and 23.6% for a detection task over 10, 20 and 50 unseen classes, compared to AP scores of 8.9%, 6.0% and 3.7% with DTW.

Our model's results are encouraging. AP results show that it is indeed feasible to perform weakly supervised detection on unseen classes using our proposed attention Siamese networks, and that future research should further develop this architecture and apply it to other domains as well, such

9

as object detection in computer vision. Though models from different domains cannot be directly compared, state-of-the-art computer vision models for object detection yield between 40 and 50 mean average precision (mAP) on the COCO dataset, for the same object classes the model was trained on (Ren et al., 2015). We expect our model's performance to increase when incorporating additional developments into the model architecture, such as pretraining the Siamese network, adding a region proposal network, adding a regression task for the prediction of more refined boundaries inside raw locations (Ren et al., 2015), and considering a few-shot learning task instead of only one-shot.

To Further investigate possible future improvements for our architecture, we compute AP for our trained Siamese attention networks using different IoU thresholds. Results are depicted in Figure 3. The results show that with the reduction of the IoU threshold, the performance of the model increases. Specifically, for the detection task over 10 unseen classes, reducing the IoU from 0.5 to 0.2 results in the AP increasing from 42.6% to 54.7%. This observation reveals that by further refining the detection location, considerable increases in performance can be gained. Such refinement can be done in future work by adding a region proposal network and a regression task for the prediction of more refined boundaries inside raw locations, as mentioned above.

5 Conclusion

We presented the attention Siamese networks model, for weakly supervised one-shot detection. The model performs a detection task over a set of unseen classes, when trained only using weak binary labels that indicate the existence of a class instance in a given example. In experiments for query-by-example spoken term detection, we outperformed a DTW baseline, while yielding a 42.6% average precision for detection across 10 unseen classes. Additional experiments show that incorporating architectural developments for more accurate bounding box predictions could result in considerable performance gains. Future work will further develop attention Siamese networks and apply them to computer vision object detection, while incorporating recent architectural developments such as a region proposal network and a regression task for the prediction of more refined boundaries inside raw locations.

Acknowledgments

References

Ao, C.-W. and Lee, H.-y. (2017). Query-by-example spoken term detection using attention-based multi-hop networks. *arXiv preprint arXiv:1709.00354*.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of International Conference on Learning Representations*, San Diago, CA.

Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., and Torr, P. H. S. (2016). Fully-convolutional siamese networks for object tracking. In *Proceedings of European conference on computer vision (ECCV) Workshops*, pages 850–865, Amsterdam, Netherlands.

Bilen, H. and Vedaldi, A. (2016). Weakly supervised deep detection networks. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 2846–2854, Las Vegas, NV.

Bromley, J., Bentz, J. W., Bottou, L., Guyon, I., LeCun, Y., Moore, C., Säckinger, E., and Shah, R. (1993). Signature verification using A "siamese" time delay neural network. *IJPRAI*, 7(4):669–688.

Chen, G., Parada, C., and Sainath, T. N. (2015). Query-by-example keyword spotting using long short-term memory networks. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5236–5240, South Brisbane, Australia.

Chorowski, J., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y. (2015). Attention-based models for speech recognition. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 577–585, Montreal, Canada.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, Miami, FL.

Dong, X., Zheng, L., Ma, F., Yang, Y., and Meng, D. (2017). Few-shot object detection. *arXiv preprint arXiv:1706.08249*.

Everingham, M., Gool, L. J. V., Williams, C. K. I., Winn, J. M., and Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338.

Eyben, F., Weninger, F., Gross, F., and Schuller, B. (2013). Recent developments in opensmile, the munich open-source multimedia feature extractor. In *Proceedings of ACM International Conference on Multimedia*, pages 835–838, Barcelona, Spain.

Girshick, R. B. (2015). Fast R-CNN. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, Santiago, Chile.

Hariharan, B. and Girshick, R. (2016). Low-shot visual recognition by shrinking and hallucinating features. *arXiv preprint arXiv:1606.02819*.

Hazen, T. J., Shen, W., and White, C. M. (2009). Query-by-example spoken term detection using phonetic posteriorgram templates. In *Proceedings of Workshop on Automatic Speech Recognition & Understanding (ASRU)*, pages 421–426, Merano/Meran, Italy.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 448–456, Lille, France.

Joder, C., Weninger, F., Wöllmer, M., and Schuller, B. (2012). The TUM cumulative dtw approach for the mediaeval 2012 spoken web search task. In *Proceedings of Proceedings of the MediaEval 2012 Workshop*.

Keren, G., Sabato, S., and Schuller, B. (2017a). The principle of logit separation. *arXiv preprint arXiv:1705.10246*.

Keren, G., Sabato, S., and Schuller, B. (2017b). Tunable sensitivity to large errors in neural network training. In *Proceedings of AAAI*, pages 2087–2093, San Francisco, CA.

Koch, G., Zemel, R., and Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, Lille, France.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Proceedings of European conference on computer vision (ECCV)*, pages 740–755, Zurich, Switzerland.

McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., and Sonderegger, M. (2017). Montreal forced aligner: trainable text-speech alignment using kaldi. In *Proceedings of INTERSPEECH*, Stockholm, Sweden.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). Librispeech: an asr corpus based on public domain audio books. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, South Brisbane, Australia.

Parada, C., Sethy, A., and Ramabhadran, B. (2009). Query-by-example spoken term detection for OOV terms. In *Proceedings of Workshop on Automatic Speech Recognition & Understanding (ASRU)*, pages 404–409, Merano/Meran, Italy.

Ren, S., He, K., Girshick, R. B., and Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 91–99, Montreal, Canada.

Teh, E. W., Rochan, M., and Wang, Y. (2016). Attention networks for weakly supervised object localization. In *Proceedings of British Machine Vision Conference 2016 (BMVC)*, York, UK.

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., and Wierstra, D. (2016). Matching networks for one shot learning. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 3630–3638, Barcelona, Spain.

Wang, C., Ren, W., Huang, K., and Tan, T. (2014). Weakly supervised object localization with latent category learning. In *Proceedings of European conference on computer vision (ECCV)*, pages 431–445, Zurich, Switzerland.

Wöllmer, M., Al-Hames, M., Eyben, F., Schuller, B., and Rigoll, G. (2009a). A multidimensional dynamic time warping algorithm for efficient multimodal fusion of asynchronous data streams. *Neurocomputing*, 73(1):366–380.

Wöllmer, M., Eyben, F., Keshet, J., Graves, A., Schuller, B., and Rigoll, G. (2009b). Robust Discriminative Keyword Spotting for Emotionally Colored Spontaneous Speech Using Bidirectional LSTM Networks. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3949–3952, Taipei, Taiwan.

Wöllmer, M., Eyben, F., Schuller, B., and Rigoll, G. (2010). Spoken Term Detection with Connectionist Temporal Classification: a Novel Hybrid CTC-DBN Decoder. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5274–5277, Dallas, TX.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 2048–2057, Lille, France.