Towards End-to-End Topic Identification on Spoken Documents without ASR (or not)

Ankita Pasad, Pedro Savarese

Toyota Technological Institute at Chicago

ankitap@ttic.edu, savarese@ttic.com

Abstract

Most of the language understanding tasks on speech data include an automatic speech recognizer (ASR) as an intermediate block. But certain high level tasks like topic identification (topic ID) do not require exceptional ASR performance because of the simple nature of the task. In recent literature a coarser preprocessing step to get token sequences from the audio without using speech transcripts or an ASR has been seen to perform well on this task. In this work we attempt to investigate whether we can get rid of these intermediate preprocessing steps and do topic ID on spoken documents in an End-to-End fashion while not using an ASR component. Our current work in progress denotes that solving this task is not straightforward, and might require large amounts of data – which we did not have in this work. While trying to circumvent the obstacles created by data scarcity, we proposed 3 different approaches for topic identification. However, our results are mostly negative in the sense that none of the proposed approaches achieved satisfying performance.

Index Terms: topic identification, convolutional neural networks, speech processing

1. Introduction

Spoken language understanding (SLU) derived applications, intent classification or topic ID for instance, on speech data are commonly solved as downstream tasks on word or subword units produced by an ASR [1]. An ASR component is used to transform a speech signal to text tokens, often using a Language Model (LM) alongside, followed by a multi-class classifier that operates on ASR output tokens. This thus requires independently training every component with non-abundant audio (learning the acoustics) and text (learning the LM) data, with supervision on multiple levels in the form of speech segmentation, transcripts for spoken utterances on word and/or phone level, and label for the class of topic or intent the spoken document belongs to. Classification-based language understanding task like topic ID does not require a perfect job at ASR [2], mainly because most of the text categorization algorithms can be seen to be operating on bag-of-words like features i.e simply accumulated tokens without any sense of order. So we need not be constrained by the accuracy of particular token instances (i.e. word error rates (WER) for instance). This could also relax the need of a strong supervision in the form of complete speech annotations if the trained deep neural network (DNN) model learns to look for the segments with relevant keywords, which is exactly what we want to explore and verify.

In this work, we have attempted to do topic ID on spoken documents without using speech transcripts. Along with the recent success of Deep Learning for solving complex tasks in an end-to-end (E2E) fashion, which is also supported by the literature - in [3] and [4], the authors use Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) respectively for end-to-end intent classification from audio - we plan to do E2E semantic understanding on speech data by doing topicID without using any supervision in the form of speech transcripts. A language understanding based end-task could also facilitate the model to do some form of semantic extraction directly on speech features.

2. Previous Work

In one of the very first attempts towards this task, Gillick et. al. [5] use a large vocabulary continuous speech recognizer on Switchboard corpus [6] and then score the hypothesis transcripts using a set of topic-specific language model. The language models are based on the automated collection of topic-specific keywords. Wintrode et. al. [2], in one of the later works, check the effect of vocabulary size and training requirement and emperically show on the Fisher Spanish Corpus [7] (40 labels) that the performance on topic ID is affected by neither high WER ($>$60%), nor by low-precision keyword-spotting ($<$40%). While the later is intuitive, as far as the recall of the model is high a low precision will not be expected to affect the performance, the former observation is the result of the way in which they are constraining the vocabulary size, i.e they are retaining the most discriminative words which would in fact aid the most in the task of topic ID.

A very recent work by Liu et. al. [8] approaches a topic ID task very similar to ours, there is only one such paper attempting to solve topic ID without ASR to the best of our knowledge. They use unsupervised solutions to automatically discover word-like, using unsupervised term discovery [9] (UTD), or phone-like, using acoustic unit discovery [10] (AUD), tokens in a speech signal and convert each spoken document into its bag-of-words representation, which is then used as a feature vector for SVM-based classification. A CNN-based framework is also used for classification on phone-like tokens since AUD-based tokenization enables a full-coverage of continuous speech into a sequence of acoustic units. UTD uses segmental dynamic time warping based algorithm to identify and cluster repeating word-like units (0.5 - 1 second) across speech, whereas AUD learns phoneme-like units from untranscribed speech using unsupervised training of hidden Markov model. The claim here is that no knowledge about the language under consideration is used other than the topic annotations. Having said that, the multilingual bottleneck features used by their model, which are based off from their previous work [11], are trained on 100 hours transcribed speech in languages other than English. This combined with specific unsupervised algorithms for tokeniza-

tion are feature-engineering heavy parts. Also, a part of speech transcripts are used for rescoring the edge weights in the acoustic similarity graph in the method of UTD. Since we plan to use a DNN framework, AUD followed by CNN-based classification is a method most similar in the flavour. Their models give impressive accuracy of 76% on topic ID, but since our aim is to explore the feasibility of solving the same task using the model which is E2E and unsupervised in complete sense, we do not use any of the feature engineering techniques and no labelled speech data for any language. Apart from zero requirement of labelled data, it will also be easier to extend our DNN model for supervised data, training on multitask loss for example, because of its E2E nature; it is not very straightforward as to how would we do it for [8].

Data-specific details are discussed in section 3. In section 4 we motivate and discuss in detail different CNN-based classifier frameworks we tried out, followed by discussion on feature extraction, experimentation details, and results in section 5. Then we conclude in section 6 while mentioning the miscellaneous experiments we tried and discuss reasonable future directions in 7

3. Dataset Statistics

We use Switchboard dataset [6] which is a telephone speech corpus. For each recording two speakers were given a predefined topic - some example topics are: recycling, capital punishment, drug testing, family finance, job benefits - and they were expected to have a conversation for a few minutes (3-5). Since this a spontaneous speech corpus it has many disfluencies (false starts, repeat starts) and "filler" utterances ("yeah", "uhh hmm", "I agree") which do not have any semantic information regarding the topic label. So the dataset characteristics gives additional reasons as to why trying to get good ASR performance would be an overkill for topic ID.

We use 30 hrs of audio data, which has the same set of conversations and labels as the baseline work [12] uses. Every conversation is split into the individual speaker sides, this creates audios with long silence regions. In order to deal with this we use weak supervision for the speech activity detection task by using the manual segmentations provided in the Switchboard corpus. The detailed statistics can be seen in table 3 under the row 'dataset A' (and B for the complete dataset). We thus have 100 conversations, i.e. 100 data points per label. Of these, 80 are used for training and the rest are divided equally between validation and test sets.

We also do some experiments on the utterance-level classification task (details in section 4.3), where each utterance is associated with the label corresponding to the topic of the conversation. For this section a stronger supervision is used in the form of utterance boundaries. An utterance is a speech segment delimited by the change of speaker in a conversation. Only utterances greater than 5 seconds were retained as we want only those speech segments which have some semantic information pertaining to the topic label. The detailed statistics can be seen in table 3 under the row 'dataset C' (and D for the complete dataset).

Feature extraction was performed using an open-source python implementation [13]. The 13-dimensional mel filtered cepstral coefficients computed over 30 milliseconds frame with 20 milliseconds overlap between consecutive frames were normalized for mean and variance.

	# of audio files	# of labels	# of hours
Dataset A	600	6	30
Dataset B	3,818	37	202
Dataset C	10,206	6	24
Dataset D	66,619	37	140

Table 1: Dataset statistics; A,B: conversation-level, C,D: utterance-level

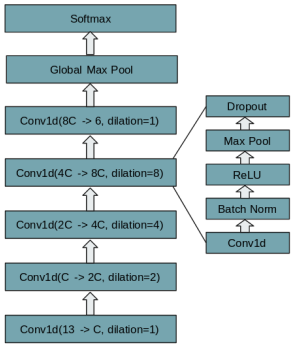

Figure 1: WaveNet-CNN

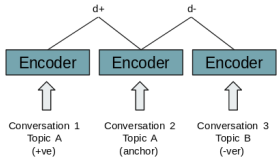

Figure 2: Triplet Network

4. Models

CNNs have proved to be extremely successful in classification tasks, not only in Computer Vision but also in NLP [14] [15]. The major advantage of CNNs over RNNs is in terms of execution time, since CNNs can exploit GPU cores to compute convolutions over sequences in parallel.

4.1. WaveNet-CNN

Since our sequences are notably long, we exploit dilated convolutions to effectively increase the receptive field of our model. WaveNet [16], a CNN especially designed for speech data, showed that a scheme of exponentially increasing dilation works well for long sequences for tasks that require large receptive fields, and thus we incorporate the same scheme in our CNN classifier. The receptive field of the proposed architecture is about 120 frames (1.2 seconds).

Our CNN is composed of 4 1D Convolutions that act as feature extractors. Each convolution is followed by Batch Norm [17], ReLU, Max Pool and Dropout [18], where the Max Pool has a window size of 2. As standard in the literature, we double the number of channels at each convolution.

Finally, our CNN contains a last convolution with 6 output channels, directly followed by a Max Pool over the entire sequence (in the temporal dimension), thus generating a 6-dimensional vector. The label probabilities are then computed by applying a Softmax on this vector. Moreover, the lack of fully-connected layers (which is possible because of the Max Pool over the entire temporal dimension) enables us to feed sequences of arbitrary length to our CNN. Figure 1 illustrates our model.

4.2. Triplet/Matching Network

Performing classification on the conversation level means that our training data is limited to only 480 points. The large discrepancy between the data dimensionality ($\approx 13 \times 60,000$) and the number of data points suggests that even small networks could easily memorize the data, hence not learning meaningful

patterns and representations.

A family of network models that proved to be useful in such regimes are Siamese [19], Triplet [20] and Matching Networks [21]. Instead of extracting high-dimensional and discriminative features from each data point, the aim of these networks is to learn a **low-dimensional metric space**: a mapping from data points to low-dimensional vectors such that similar points are close (e.g. have small distance) under this mapping. Formally (for Triplet Networks), we want to learn a mapping $f : \mathcal{X} \rightarrow \mathbb{R}^d$ such that $\forall(x_i, x_j, x_k \in S^3 : y_i = y_j \neq y_k) \quad d(f(x_i), f(x_j)) < d(f(x_i), f(x_k))$, where $d(\cdot)$ is a distance metric (in general, the Euclidean distance). Note that learning such mapping requires satisfying $O(|S|^3)$ constraints, which should provide more (much needed) supervision when training the network.

In order to learn f through a neural network, define a model E called the *encoder*, which maps each data point to a low-dimensional vector. Given 3 data points x_i, x_j, x_k with $y_i = y_k \neq y_k$, we first map (encode) the three data points to the low-dimensional space: $e_i = E(x_i), e_j = E(x_j), e_k = E(x_k)$. Then, we compute the distances $d^+ = d(e_i, e_j), d^- = d(e_i, e_k)$ (Figure 2), and minimize the loss $l(x_i, x_j, x_k) = \left[m - d^+ + d^-\right]_+$, where m is the margin (we use $m = 0.02$). During training, we iterate over each conversation $x_i \in S$, and uniformly sample conversations x_j, x_k s.t. $y_i = y_j \neq y_k$. With this, we define each epoch as iterating over each conversation – that is, a total of $|S|$ parameter updates.

Finally, for inference we use a method inspired by Matching Networks [21]: given a test point x, we find the point in the training set S with smallest distance to it (under the learned mapping), and use its label as prediction. Formally, $\hat{y} = y_{i^\star}$, where $x_{i^\star} = argmin_{x'} d(x, x')$. This is equivalent to performing 1-Nearest Neighbor in the learned space.

4.3. Global/Local Loss

Even a Triplet Network might not be enough for such a tiny dataset when we consider each conversation as being one data point. One alternative is to perform utterance-wise classification: for each point (x, y) in our training set S, we split x into utterances $u_1, \ldots, u_k$, and train a CNN to classify each of them as y: that is, the correspondent loss for x is $\frac{1}{k}\sum_{i=1}^{k} l(u_i, y)$. For this task, we use Dataset C described in section 3. As for the CNN, we use the WaveNet-CNN described previously.

While a larger quantity of training examples should facilitate our model to learn meaningful representations instead of purely memorizing conversations, this causes a few issues that must be properly addressed.

First, even after removing silence and short utterances, it is overly optimistic to expect that most utterances contain enough information to predict the topic of the full conversation, and hence we must find a way to ignore poor (uncertain) predictions while considering good (confident) ones. Second, it is unclear how inference ought to be performed: the end goal is to predict topics for **conversations** in the test set, and not for individual utterances: therefore, we must design a way to also perform conversation-wise prediction.

We propose a Global/Local training scheme, where by "local" we denote utterance-wise classification as described above. Additionally, addressing both mentioned issues, we perform *utterance-wise max pooling* on the outputs of the WaveNet-CNN. More specifically, given utterances $u_1, \ldots, u_k$, we pass them through the CNN, which in turn outputs scores $s_1, \ldots, s_k$, $s_i \in \mathbb{R}^6$ (which are used to compute the local loss, after a Softmax layer). Next, we compute $s' \in \mathbb{R}^6$ where $s'(i) = \max\{s_1(i), \ldots, s_k(i)\}$, where $s'(i)$ is the i'th component of the s' vector. This is equivalent to a channel-wise Max Pool over utterances, and s' will contain the largest scores for each topic, accross utterances. With this, we can use a Softmax layer to compute conversation-wise topic probabilities, and define a "global" loss $l(s', y)$. The final loss used for training is the sum of the local and global losses, as in Figure 4.3.

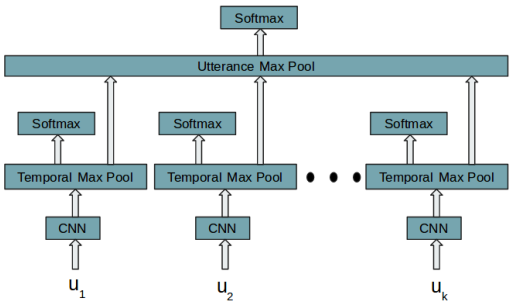

Figure 3: Global/Local scheme for training a classifier on both utterance and conversation levels.

5. Experimentation

5.1. Conversation-wise Classification

Our first set of experiments involve training WaveNet-CNNs to perform conversation-wise classification. Optimization is done using cross-entropy loss.

We start by exploring architecture settings, such as kernel size for the convolutional layers and number of channels. Since we double the number of channels after each pooling layer, the number of channels per convolution in the WaveNet-CNN is $C, 2C, 4C, 8C, 6$ (the number of channels in the last convolution equals the number of classes, since each channel will contain label probabilities afterwards), where C is the number of channels of the first convolutional layer.

First, we try different values for the base channel size $C \in \{4, 8, 16, 32\}$, to find a network suitable for the task. We set the kernel size to 5, and train the 4 networks with the Adam optimizer (using its default parameters). We observe that all 4 networks overfit on the training set – even the smallest network, which contains only $4,700$ parameters. We also tried kernel sizes $\{3, 7, 11\}$, and did not note much difference from setting it as 5.

Next, we investigated whether L_2 regularization and Dropout could stop the network from overfitting. We found out (Figure 4) that even with reasonable regularization (L_2 regularization $\lambda = 0.1$ and Dropout rate $p = 0.2$ per layer), the network with $C = 8$ still fails to generalize.

We also experimented with different learning rates for Adam (0.1, 0.01, 0.001, 0.0001), again not observing improvements on generalization.

Next, we proceed to evaluate whether the Triplet Network manages to achieve better generalization. Since they usually require considerably more iterations to be properly trained (there are more constraints and possible inputs), we decrease the input lengths by first only using the first 15,000 samples of each conversation, and next replacing each block of 5 consecutive frames by its average. This effectively decreases the input length to

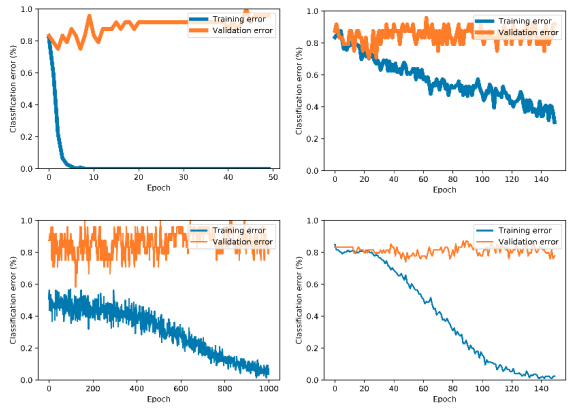

Figure 4: Top left: training plot for WaveNet-CNN with $C = 8$ and no regularization. Top right: training plot for WaveNet-CNN with $C = 8$, L_2 regularization $\lambda = 0.1$ and dropout rate $p = 0.2$. Bottom left: training plot for Triplet Network with L_2 regularization 0.1 and Dropout rate 0.2.

	WaveNet-CNN	TripletNet	Global/Local
Test err.	79.1%	83%	78%

Table 2: Test errors of different approaches: WaveNet-CNN, Triplet Network and Global/Local WaveNet-CNN classification.

3,000 frames.

Additionally, for the encoder we use 4 convolutions total, instead of 5 as in the WaveNet-CNN, and generated embeddings have size 100. Figure 4 shows that Triplet Networks still fail to generalize, even with regularization ($\lambda = 0.1, p = 0.2$). Again, trying different hyperparameters ($p \in \{0.3, 0.4\}$, $C \in \{8, 16\}$) did not provide any observable increase in validation performance.

5.2. Utterance-wise Classification

Finally, we try using utterances instead of whole conversations for classification, with the hope that the increased number of training examples helps against overfitting.

We use a WaveNet-CNN with 4 convolutions followed by a Max Pool over the entire sequence as our utterance classifier. That is, given an utterance, it generates a 6-dimensional output of (unnormalized) scores. The local loss is the average (over utterances) of the cross-entropy loss applied to each utterance, after an additional Softmax layer on the scores.

For the global loss, we compute a max pool **over utterances** on the score vectors. That is, given a matrix $k \times 6$, where k is the number of utterances, we perform a row-wise max operation, generating a final 6 dimensional vector, which we pass through a Softmax to generate conversation-level label probabilities. The global loss is computed from this probability vector, and summed with the local loss.

Once again, we observe that even though the network could not quickly memorize the training data, the results on the validation set were still poor (Figure 4) even with regularization.

The test performances for each of the approaches can be found on Table 2. Note that a random classififer would achieve 83.3% test error.

6. Discussion

From the nature of the results obtained it is evident that the phenomenon of over-fitting is the real culprit here and even if we add lots of regularization the model generalization capability is really low. In order to verify whether the data size alone an issue we trained the network on complete switchboard dataset (Dataset B and D from section 3 for the conversation-level and utterance-level models respectively), while the chance accuracy is $\approx 2\%$, the best accuracy model could get to was $\approx 6\%$.

From the analysis point-of-view, we added an attention layer at the input in the very first model (section 4.1) in order to visualize the regions in an audio segments which are given higher weightage in the decision making process. After a few epochs the attention weights converged to just a single frame while the model was giving close to 100% accuracy. We also did a "cheating" experiment just to check how much would the availability of transcripts help. Using the simple bag-of-words features for the ground truth transcriptions at the conversation level and training a 2 layer neural network with sigmoid non-linearity, the model gives 100% train and dev accuracy on Dataset A, and 100% train and 88% dev accuracy on Dataset B. Thus showing that having transcripts would definitely help, as expected.

Thus it is clear that the kind of models we are using are not best suited for the task and we need novelty either in terms of input features or the loss we are optimizing for or the model architecture. Based on these observations we have a couple of reasonable future directions described in the next section.

7. Future Work

In the presence of a preprocessing step which enables a full-coverage of continuous speech into a sequence of acoustic units, it is very common to do keyword spotting-like task to discover discriminative units for each topic and to transform the feature vector accordingly before putting it through the classification framework. Scaling the feature vector by inverse document frequency (IDF) to produce TF-IDF features is one such technique [8]. Similarly for our current task, we can perform clustering over the final output vectors (embeddings) of each conversation input for the whole train corpus. The cluster identifiers of the embeddings of a conversations will then serve as the terms and the clusters themselves will be documents to transform the features to TF-IDF features. This way we will also have lower dimensional and more meaningful feature space to perform the classification task on.

Another idea involves including supervision into the model but focuses on the task of semantic information extraction from speech. Given the experimental results it seems like attempting to solve this task in a completely unsupervised fashion is a very challenging task. So one thing that would make sense is to train a multi-task framework with ASR and topic ID being 2 tasks. As we saw in [2] a perfect ASR is not necessary for a good performance on topic ID. So such a framework would help us find an ASR regime optimal enough to get a good topic ID performance and also check if having a topic ID loss helps ASR learn to predict only semantically relevant words.

8. Acknowledgements

We would like to thank Prof. Karen Livescu, Shubham Toshniwal and Shane Settle for their suggestions and involvement throughout the course of the project.

9. References

[1] T. J. Hazen, "Mce training techniques for topic identification of spoken audio documents," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 2451–2460, 2011.

[2] J. Wintrode and S. Khudanpur, "Limited resource term detection for effective topic identification of speech," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014.

[3] Y.-P. Chen and R. Price, "Spoken language understanding without speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018.

[4] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018.

[5] L. Gillick, J. Baker, J. Baker, J. Bridle, M. Hunt, Y. Ito, S. Lowe, J. Orloff, B. Peskin, R. Roth *et al.*, "Application of large vocabulary continuous speech recognition to topic and speaker identification using telephone speech," in *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*. IEEE, 1993.

[6] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, vol. 1. IEEE, 1992, pp. 517–520.

[7] C. Cieri, D. Miller, and K. Walker, "The fisher corpus: a resource for the next generations of speech-to-text," in *Language Resources and Evaluation Conference*, 2004.

[8] C. Liu, J. Trmal, M. Wiesner, C. Harman, and S. Khudanpur, "Topic identification for speech without asr," *arXiv preprint arXiv:1703.07476*, 2017.

[9] A. Jansen and B. Van Durme, "Efficient spoken term discovery using randomized algorithms," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011.

[10] L. Ondel, L. Burget, and J. Černockỳ, "Variational inference for acoustic unit discovery," *Procedia Computer Science*, vol. 81, pp. 80–86, 2016.

[11] C. Liu, J. Yang, M. Sun, S. Kesiraju, A. Rott, L. Ondel, P. Ghahremani, N. Dehak, L. Burget, and S. Khudanpur, "An empirical evaluation of zero resource acoustic unit discovery," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5305–5309.

[12] C. Liu, "Baseline Switchboard data split," https://github.com/cliu1/lorelei-amdtk/tree/master/recipes/swbd/data, 2017.

[13] J. Lyons, "Python Speech Features," https://github.com/jameslyons/python_speech_features, 2017.

[14] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. van den Oord, A. Graves, and K. Kavukcuoglu, "Neural Machine Translation in Linear Time," *ArXiv e-prints*, 2016.

[15] Y. Zhang and B. Wallace, "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification," *ArXiv e-prints*, 2015.

[16] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," *ArXiv e-prints*, 2016.

[17] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *ArXiv e-prints*, 2015.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, 2014.

[19] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," 2015.

[20] E. Hoffer and N. Ailon, "Deep metric learning using Triplet network," *ArXiv e-prints*, 2014.

[21] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching Networks for One Shot Learning," *ArXiv e-prints*, 2016.