INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Attention-Based Keyword Localisation in Speech using Visual Grounding

Kayode Olaleye, Herman Kamper

Department of E&E Engineering, Stellenbosch University, South Africa
kaykola.olaleye@gmail.com, kamperh@sun.ac.za

Abstract

Visually grounded speech models learn from images paired with spoken captions. By tagging images with soft text labels using a trained visual classifier with a fixed vocabulary, previous work has shown that it is possible to train a model that can *detect* whether a particular text keyword occurs in speech utterances or not. Here we investigate whether visually grounded speech models can also do keyword *localisation*: predicting where, within an utterance, a given textual keyword occurs without any explicit text-based or alignment supervision. We specifically consider whether incorporating attention into a convolutional model is beneficial for localisation. Although absolute localisation performance with visually supervised models is still modest (compared to using unordered bag-of-word text labels for supervision), we show that attention provides a large gain in performance over previous visually grounded models. As in many other speech-image studies, we find that many of the incorrect localisations are due to semantic confusions, e.g. locating the word 'backstroke' for the query keyword 'swimming'.

Index Terms: multimodal modelling, keyword localisation, visual grounding, attention, word discovery

1. Introduction

Several studies have considered how speech processing systems can be developed in the absence of conventional transcriptions [1–6]. This could enable speech technology in low-resource settings where full transcriptions are not available. One form of weak supervision is using images paired with spoken captions [7–16]. Compared to using labelled data, this form of visual supervision is closer to the types of signals that infants would have access to while learning their first language [17–22], and to how one would teach new words to robots using spoken language [23, 24]. It is also conceivable that this type of visual supervision could be easier to obtain when developing systems for low-resource languages [25], e.g. when it is not possible to collect textual labels for a language without a written form.

In this visually supervised setting, both the utterance and the paired image are unlabelled. Some form of self-supervision is therefore required to train a model. One approach is to train a model that projects images and speech into a joint embedding space [8, 26]. Another approach is to use an external image tagger with a fixed vocabulary to obtain soft text labels, which are then used as targets for a speech network that maps speech to keyword labels [27, 28]. The advantage of this latter approach is that it can be used to *detect* whether a textual keyword is present in a new input utterance (as long as the keyword is in the vocabulary). Without seeing any transcriptions, previous work has shown that such visually grounded models can do keyword detection at a high precision [9]. However, current approaches does not indicate *where* in an utterance a detected keyword occurs. In this paper we investigate whether visually grounded speech models can be used for this task of *keyword localisation*: predicting where in an utterance a given textual keyword occurs. The model needs to do so without any explicit text labels or word position supervision.

In our own initial efforts toward this goal [29], we tried to adapt a saliency-based method often used in the vision community for determining which parts of an input image contributes most to a particular output prediction [30]. We found, however, that this approach was poorly matched to the particular multi-label loss used for the visually grounded model. More specifically, this approach gave worse localisation performance than a visually grounded version of what we call the PSC model [4] (based on the author names). The original PSC uses a convolutional neural network (CNN) architecture that is tailored to jointly locate and classify words using a bag-of-words (BoW) labelling, i.e. the model is trained with labels indicating the presence or absence of a word but not the location, order, or number of occurrences. We adapted this model to use visual supervision. Although the visually grounded PSC model gave better localisation scores relative to the saliency-based approach, absolute performance was still low.

In this paper we specifically consider whether incorporating *attention* into a visually grounded CNN is beneficial for localisation. Attention has become a standard mechanism to allow a neural network to automatically weigh different parts of its input when producing some (intermediate) output [31, 32]. In our architecture, a given textual keyword is mapped to a learned query embedding, which is then used to weigh the relevance of intermediate convolutional filters. The model is trained using a multi-label loss using the soft tags from an external visual classifier. This architecture is heavily inspired by [33], which considered supervised keyword localisation in sign language; instead of using text labels, here we use visual grounding.

Without using any explicit text or alignment supervision, we show that an attention-based model outperforms the visually grounded PSC by roughly 17% absolute in localisation $F1$. However, the absolute performance is still modest. We subsequently do an error analysis to determine the types of localisation mistakes that the model makes. As in previous studies on visually grounded speech models [9, 27], we find that in many cases the model is penalised for locating a word which is not an exact match to the query but is semantically related.

2. CNNs for keyword localisation in speech

2.1. Attention-based CNN for keyword localisation

Our attention-based CNN is heavily inspired by [33], which proposed an attention-based graph-convolutional network over skeleton joints to search for keywords in sign language sentences. In our case, we incorporate attention in a CNN model for localising keywords in speech, and train the model with visual supervision instead of explicit labels for whether a keyword is present in an utterance or not. Our architecture, shown in Figure 1, consists of four modules: a *convolutional module* to

Copyright © 2021 ISCA

http://dx.doi.org/10.21437/Interspeech.2021-435

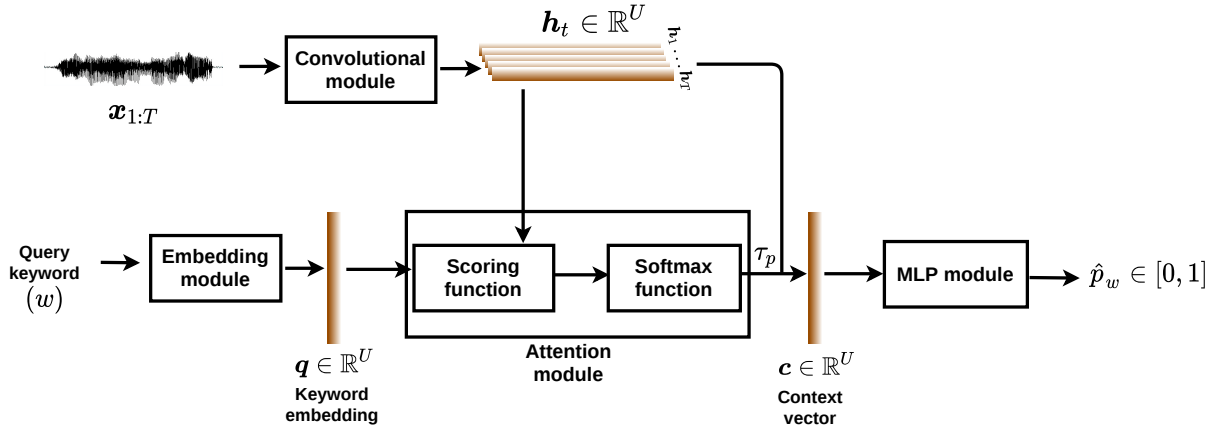

Figure 1: *The structure of our attention-based CNN for keyword localisation in speech.*

extract features from speech, an *embedding module* to represent a query keyword, an *attention module* to compress the convolutional features into a context vector containing features most relevant to reaching a decision about the query keyword, and a *multi-layer perceptron* (MLP) module that takes the output of the attention module and produces a probability about the presence or absence of the query keyword in the speech.

The convolutional module consists of a succession of one-dimensional convolutional layers that produces the intermediate features that are weighed using the query embedding vector. This module takes as input a sequence of T Mel-frequency cepstral coefficient vectors $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$ and outputs a sequence of convolutional features $\boldsymbol{h}_{1:T}$, each of dimension U.

The embedding module takes a text keyword w as input and maps it to a U-dimensional query vector $\boldsymbol{q}$ using an index-based lookup table. U is selected to match the dimension of the sequence of convolutional features $\boldsymbol{h}_{1:T}$. The type of keyword needs to be in the vocabulary of the system, and the embedding matrix is trained jointly with the rest of the model.

The attention module [31, 34] starts by computing a score e_t for each time step t by using a dot product to measure the similarity between each time step along the convolutional features $\boldsymbol{h}_t$ and the query embedding: $e_t = \boldsymbol{q} \cdot \boldsymbol{h}_t$. The similarity score e_t is then converted into an attention weight α_t over time steps using a softmax function: $\alpha_t = \frac{\exp(e_t)}{\sum_{t'=1}^{T} \exp(e_{t'})}$. The attention weight is used to compute the context vector $\boldsymbol{c}$ by weighing the features $\boldsymbol{h}_t$ at every time step: $\boldsymbol{c} = \sum_{t=1}^{T} \alpha_t \boldsymbol{h}_t$.

Finally, the MLP module consumes $\boldsymbol{c}$, passes it through a number of fully-connected layers, and terminates with a layer with a sigmoid activation producing a single probability output indicating the presence or absence of the keyword, i.e. $\hat{p}_w = \sigma(\text{MLP}(\boldsymbol{c}))$ and we interpret $\hat{p}_w \in [0, 1]$ as $P(w|\boldsymbol{X})$.

How is the model trained? Let's say we know at training time whether a particular keyword w occurs in an output utterance $\boldsymbol{X}$, i.e. we have an indicator variable $y_{\text{bow},w} \in \{0, 1\}$. (But we still don't know where w occurs). In this case we could train the model with the binary cross-entropy loss:

$$J(\hat{p}_w, y_{\text{bow},w}) = y_{\text{bow},w} \log \hat{p}_w + (1 - y_{\text{bow},w}) \log(1 - \log \hat{p}_w)$$

where $\hat{p}_w$ is the output of the model for a single training example $\boldsymbol{X}$ and we look up the w^{th} embedding vector $\boldsymbol{q}$ for the attention model. When training with paired images and spoken captions, we don't actually know whether w occurs in the training utterance $\boldsymbol{X}$. So we use the above loss but replace $y_{\text{bow},w}$ with $y_{\text{vis},w} \in [0, 1]$, the soft probability obtained for keyword w by passing the paired image through an external visual tagger.

How can the model detect whether a keyword is present or absent in a new test utterance? At test time, a threshold θ can be applied to the output of the model $\hat{p}_w$ to determine the presence or absence of the query keyword w. How can the model locate where a detected keyword occurs? At test time, we select the position of the highest attention weight α_t produced by the model as the predicted location τ_p of a detected keyword w.

2.2. Model variants

For the convolutional module, we consider two different structures. The first, based on [4], consist of six convolutional layers without any intermediate max-pooling. We refer to the resulting model that uses this convolutional module together with the query and attention modules as *CNN-Attend*. The second structure, based on [9], uses three convolutional layers with intermediate max-pooling. We refer to the resulting model as *CNN-PoolAttend*.

2.3. Baseline models

The two architectures for the convolutional module above are actually based on two non-attention CNN models from previous work, which also serves as our baselines. We refer to the first as PSC [4], which is the non-attention parallel to the CNN-Attend model. PSC was originally designed to jointly locate and classify words using only BoW supervision. It operates on $\boldsymbol{X}$ using a stack of convolutional layers and an aggregation function. It performs keyword localisation using the output of its last convolutional layer $\boldsymbol{h}_{1:T}$. Here, each $\boldsymbol{h}_t$ is a W-dimensional vector, with $h_{t,w} = \alpha_{t,w}$ giving the score for word w at time t. W is the number of words in the vocabulary. A single utterance-level detection score is obtained by feeding the frame-level scores into the aggregation function: $g_w(\boldsymbol{X}) = \frac{1}{r} \log \left[\frac{1}{T} \sum_{t=1}^{T} \exp \{r h_{t,w}(\boldsymbol{X})\} \right]$. Finally, a sigmoid function converts the speech-level scores into a probability $\hat{\boldsymbol{y}} = \sigma(\boldsymbol{g}(\boldsymbol{X}))$, with each $\hat{y}_w$ giving the probability that word w occurs in $\boldsymbol{X}$. The model is trained with the cross-entropy loss:

$$J(\hat{\boldsymbol{y}}, \boldsymbol{y}_{\text{bow}}) = - \sum_{w=1}^{W} \{ y_{\text{bow},w} \log \hat{y}_w + (1 - y_{\text{bow},w}) \log(1 - \log \hat{y}_w) \} \quad (1)$$

where $\boldsymbol{y}_{\text{bow}} \in \{0,1\}^W$ indicates that BoW supervision is used. For visual supervision, it is replaced by $\boldsymbol{y}_{\text{vis}} \in [0,1]^W$, which is the vector of soft probabilities for the W words in the output of the multi-label visual tagger.

Our second non-attention baseline is *CNN-Pool*, the parallel to *CNN-PoolAttend*. It uses a number of convolutional layers with intermediate max-pooling, max-pooling over time over its last convolutional layer, and a number of fully connected layers before terminating in a sigmoid output. The output is again denoted as $\hat{\boldsymbol{y}}$ and the same loss as in (1) is used. While PSC has a localisation mechanism built into the architecture, this is not the case for CNN-Pool. As in [29], we therefore use the GradCAM [30] saliency-based localisation method to do localisation. In short, this works by first determining how important each filter in the output of the last one-dimensional convolutional layer $\boldsymbol{h}_{1:T}$ is to the word w, given by $\gamma_{k,w} = \frac{1}{T}\sum_{t=1}^{T}\frac{\partial \hat{y}_w}{\partial h_{t,k}}$. Each filter is then weighed by its importance, giving the localisation scores $\alpha_{t,w} = \text{ReLU}\left[\sum_{k=1}^{K}\gamma_{k,w}h_{t,k}\right]$.

3. Experimental setup

3.1. Data

We perform experiments on the Flickr8k Audio Caption Corpus of [7], consisting of five English audio captions for each of the roughly 8k images. A split of 30k, 5k and 5k utterances are used for training, development and testing, respectively.

The visually grounded models are trained using soft text labels obtained by passing each training image through the multi-label visual tagger of [27], which is based on VGG-16 [35] and is trained on images disjoint from the data used here [36–38]. This tagger has an output of 1000 image classes, but here we use a system vocabulary corresponding to $W = 67$ unique keyword types. This set of keywords is the same set used in [27, 39], and includes words such as 'children', 'young', 'swimming', 'snowy' and 'riding'. The procedure used to select these keywords are detailed in [27]; it includes a human reviewer agreement step, which reduced the original set from 70 to 67 words.

3.2. Model architectures

The $W = 67$ soft probabilities from the tagger are used as the targets $\boldsymbol{y}_{\text{vis}}$ for the visually grounded models. As an upper limit for how well we can expect these visual models to perform, we train models with similar structures but using BoW labels $\boldsymbol{y}_{\text{bow}}$ obtained using transcriptions for the same set of keywords.

We first look at the baseline models (Section 2.3). The PSC model consists of six one-dimensional convolutional layers with ReLU activations. The first has 96 filters with a kernel width of 9 frames. The next four has a width of 11 units with 96 filters. The last convolutional layer, with 1000 filters and a width of 11 units, is fed into the aggregation function with a final sigmoid activation. The CNN-Pool model consists of three one-dimensional convolutional layers with ReLU activations. Intermediate max-pooling over 3 units are applied in the first two layers. The first convolutional layer has 64 filters with a width of 9 frames. The second layer has 256 filters with a width of 11 units, and the last layer has 1024 filters with a width of 11. Global max-pooling is applied followed by two fully connected layers each with 4096 and 1000 filters respectively and terminates in sigmoid activation to obtain the final output for the $W = 67$ words.

Next we turn to our attention-based models (Section 2.2). The convolutional module of CNN-Attend has the same structure as the PSC model above while the convolutional module of CNN-PoolAttend has the same structure as CNN-Pool. The embedding module embeds input query keywords into $U = 1000$ dimensional embeddings in CNN-Attend and $U = 1024$ dimensional embeddings in CNN-PoolAttend. For both models, the MLP module consists of a fully connected ReLU layer with 4096 units terminating in a sigmoid layer giving the single probability score as the final model output.

All models are implemented in PyTorch and uses Adam optimisation [40] with a learning rate of $1 \cdot 10^{-4}$.

3.3. Evaluation

We evaluate localisation performance for a query keyword w as follows. We first check whether the detection score ($\hat{p}_w$ or $\hat{y}_w$, depending on the model in Section 2) of w is greater or equal to a threshold θ, and then select the position of the highest attention weight (α_t or $\alpha_{t,w}$) as the predicted location τ_p of w. τ_p is accepted as the correct location of a word if it falls within the interval corresponding to the true location of the word, according to forced alignments. The model is penalised whenever it localises a word whose detection score is less than θ, i.e., if a word is not detected, then it is counted as a failed localisation even if the attention weight is at a maximum within that word. We set the value of θ to 0.4 and 0.5 for BoW trained models and visually supervised models, respectively, which gave best performance on development data. We compute precision, recall, and $F1$ scores using τ_p.[1]

4. Experimental results

Table 1 shows the keyword localisation performance for the attention-based models (CNN-Attend and CNN-PoolAttend), and the baseline models (PSC and CNN-Pool) when supervised with BoW labels and visual targets. The localisation scores are computed with detection scores taken into account (Section 3.3). The goal is to see how well attention-based models perform on

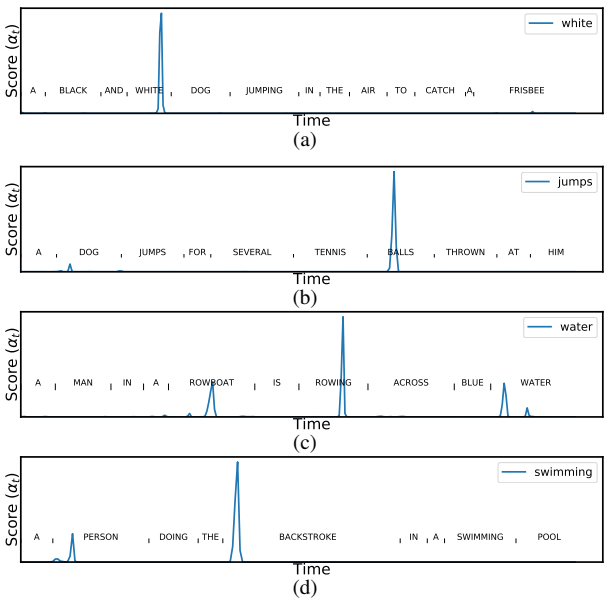

Figure 2: *Examples of localisation with the visually supervised CNN-Attend model. The query keyword is shown on the right.*

[1] In [4, 29], this is referred to as the *actual* metric.

Table 1: *Keyword localisation scores (%), where the task is to predict where an utterance a given keyword occurs.*

	Bag-of-words supervision			Visual supervision		
Model	P	R	$F1$	P	R	$F1$
Attention-based models:						
CNN-Attend	58.1	68.1	62.7	31.2	15.0	20.3
CNN-PoolAttend	30.8	47.2	37.3	10.0	8.0	8.9
Baseline models:						
PSC	78.4	70.4	74.2	7.6	2.2	3.4
CNN-Pool	10.3	32.1	15.7	7.1	10.8	8.6

Table 2: *Keyword detection scores (%), where the task is to detect whether a given keyword occurs in an utterance (regardless of location).*

	Bag-of-words supervision			Visual supervision		
Model	P	R	$F1$	P	R	$F1$
Attention-based models:						
CNN-Attend	82.8	79.2	80.9	30.6	27.2	28.8
CNN-PoolAttend	74.0	70.6	72.3	18.6	23.0	20.5
Baseline models:						
PSC	86.9	70.4	77.8	20.8	7.0	10.4
CNN-Pool	81.5	73.0	77.0	33.1	26.2	29.2

the keyword localisation task relative to models without attention. CNN-Attend trained with visual supervision achieves a localisation precision of 31.2% and an $F1$ of 20.3%, outperforming the visually supervised PSC model. However, we observe a contrasting results on the BoW-supervised models, where performance actually gets worse when comparing the PSC and CNN-Attend models. Comparing the BoW-supervised CNN-PoolAttend and CNN-Pool, we do see that attention gives better localisation than the GradCAM saliency-based localisation, but both of these models are outperformed by the PSC architecture.

The one reason that we did consider the convolutional architecture in CNN-PoolAttend and CNN-Pool (Section 2.2) is that it gives better scores than the other structure when only considering keyword detection, the task of predicting whether a keyword is present or not, irrespective of the localisation. This is shown in Table 2. But here we see that the visual CNN-Attend achieves close to the detection $F1$ of CNN-Pool (28.8% vs 29.2%) with the benefit of dramatically better localisation (20.3% vs 8.6%). Although absolute performance is still modest, the performance of the visually trained CNN-Attend suggests that attention-based CNN models is a better model structure for keyword localisation when hard ground truth labels are not available.

To better understand the failure modes of the best visually supervised model, CNN-Attend, Figure 2 shows localisation examples on test data. Figure 2(a) is an example of a case where the visually trained model successfully localises 'white' although it is trained without explicit location information and hard ground truth text targets. Figures 2(b), (c) and (d) are examples of failure cases. Figure 2(b) shows an outright failure. Figure 2(c) shows that when the model is prompted with 'water', the attention weight is high on 'rowing', 'rowboat' and 'water', with 'rowing' assigned the highest localisation score. All of these words are semantically related to the query word. Figure 2(d) shows that the model also sometimes localises a single semantically related word in the utterance: the keyword 'backstroke' is assigned highest localisation score when prompted with 'swimming'.

To roughly quantify the different types of errors (of the types depicted in Figure 2), we randomly sample 100 test utterances and visualise where the model attempts to locate a randomly selected keyword among the 67 keywords in the system vocabulary. We then categorise the 100 visualisations into four groups. We find that, in this small sample set, the model makes 48% correct localisations, 39% outright incorrect localisations, 7% single-word semantic localisations, and 6% multi-word semantic localisations, where the attention fires on the target word but also on multiple other words as in Figure 2(d). The outright mistakes are therefore still high, but roughly 13% of the mistakes can be seen as semantically related.

5. Conclusions

We investigated whether keyword localisation in speech is possible with attention-based convolutional neural networks (CNNs) trained with visual context. We specifically compared two CNN models with attention to two non-attention baseline models used in previous work. We found that adding attention improved localisation performance, but scores are still far from models trained with an unordered bag-of-word supervision (using transcriptions). We qualitatively showed that the visually trained models sometimes locate semantically related words. Future work can examine cases where the models are evaluated not just on their ability to propose location of the exact keyword of interest, but also words that are semantically related to the keyword of interest. Apart from our own preliminary work [29], the only other work that considered localisation with a visually supervised model is [11, 41]. But their localisation at test time is based on having an image containing the word of interest (a text keyword is not provided). Future work could also see if insights from these two settings can be combined.

Acknowledgements. This work is supported in part by the National Research Foundation of South Africa (grant no. 120409), a Google Faculty Award for HK, and a Google Africa PhD Scholarship for KO. We would like to thank Benjamin van Niekerk for helpful input.

6. References

[1] J. Driesen, H. Van hamme, and W. B. Kleijn, "Learning from images and speech with non-negative matrix factorization enhanced by input space scaling," in *Proc. SLT*, 2010.

[2] G. Synnaeve, M. Versteegh, and E. Dupoux, "Learning words from images and speech," *NeurIPS Workshop Learn*, 2014.

[3] L. Duong, A. Anastasopoulos, D. Chiang, S. Bird, and T. Cohn, "An attentional model for speech translation without transcription," in *Proc. NAACL-HLT*, 2016.

[4] D. Palaz, G. Synnaeve, and R. Collobert, "Jointly learning to locate and classify words using convolutional networks," in *Proc. Interspeech*, 2016.

[5] S. Settle, K. Levin, H. Kamper, and K. Livescu, "Query-by-example search with discriminative neural acoustic word embeddings," in *Proc. Interspeech*, 2017.

[6] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," in *Proc. Interspeech*, 2017.

[7] D. Harwath and J. Glass, "Deep multimodal semantic embeddings for speech and images," in *Proc. ASRU*, 2015.

[8] D. Harwath, A. Torralba, and J. Glass, "Unsupervised learning of spoken language with visual context," in *Proc. NeurIPS*, 2016.

[9] H. Kamper, S. Settle, G. Shakhnarovich, and K. Livescu, "Visually grounded learning of keyword prediction from untranscribed speech," in *Proc. Interspeech*, 2017.

[10] D. Harwath, A. Torralba, and J. R. Glass, "Unsupervised learning of spoken language with visual context," in *Proc. NeurIPS*, 2016.

[11] D. Harwath, A. Recasens, D. Surís, G. Chuang, A. Torralba, and J. Glass, "Jointly discovering visual objects and spoken words from raw sensory input," in *Proc. ECCV*, 2018.

[12] H. Kamper and M. Roth, "Visually grounded cross-lingual keyword spotting in speech," in *Proc. SLTU*, 2018.

[13] D. Harwath, G. Chuang, and J. Glass, "Vision as an interlingua: Learning multilingual semantic embeddings of untranscribed speech," in *Proc. ICASSP*, 2018.

[14] R. Eloff, H. A. Engelbrecht, and H. Kamper, "Multimodal one-shot learning of speech and images," in *Proc. ICASSP*, 2019.

[15] D. Harwath and J. Glass, "Towards visually grounded sub-word speech unit discovery," in *Proc. ICASSP*, 2019.

[16] D. Harwath, W.-N. Hsu, and J. Glass, "Learning hierarchical discrete linguistic units from visually-grounded speech," in *Proc. ICLR*, 2019.

[17] P. C. Bomba and E. R. Siqueland, "The nature and structure of infant form categories," *J. Exp. Child Psychol.*, 1983.

[18] S. Pinker, "The language instinct," *Harper Perennial, New York*, 1994.

[19] P. D. Eimas and P. C. Quinn, "Studies on the formation of perceptually based basic-level categories in young infants," *Child Develop.*, 1994.

[20] D. Roy, "Grounded spoken language acquisition: experiments in word learning," *IEEE Trans. Multimedia*, 2003.

[21] L. Boves, L. ten Bosch, and R. Moore, "ACORNS – towards computational modeling of communication and recognition skills," in *Proc. ICCI*, 2007.

[22] L. Gelderloos and G. Chrupała, "From phonemes to images: levels of representation in a recurrent neural model of visually-grounded language learning"," in *Proc. COLING*, 2016.

[23] M. Sun and H. Van hamme, "Joint training of non-negative tucker decomposition and discrete density hidden markov models," *Comput. Speech Lang.*, 2013.

[24] L. Nortje and H. Kamper, "Unsupervised vs. transfer learning for multimodal one-shot matching of speech and images," in *Proc. Interspeech*, 2020.

[25] V. De Sa and D. Ballard, "Category learning through multimodality sensing," *Neural Comput.*, 1998.

[26] G. Chrupała, L. Gelderloos, and A. Alishahi, "Representations of language in a model of visually grounded speech signal," in *Proc. ACL*, 2017.

[27] H. Kamper, G. Shakhnarovich, and K. Livescu, "Semantic speech retrieval with a visually grounded model of untranscribed speech," *IEEE/ACM Trans. Acoust., Speech, Signal Process.*, 2019.

[28] A. Pasad, B. Shi, H. Kamper, and K. Livescu, "On the contributions of visual and textual supervision in low-resource semantic speech retrieval," in *Proc. Interspeech*, 2019.

[29] K. Olaleye, B. van Niekerk, and H. Kamper, "Towards localisation of keywords in speech using weak supervision," *SAS NeurIPS Workshop*, 2020.

[30] R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," in *Proc. ICCV*, 2017.

[31] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. EMNLP*, 2015.

[32] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016.

[33] N. C. Tamer and M. Saraçlar, "Keyword search for sign language," in *Proc. ICASSP*, 2020.

[34] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. CVPR*, 2009.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NeurIPS*, 2012.

[38] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. CVPR*, 2014.

[39] H. Kamper, A. Anastassiou, and K. Livescu, "Semantic query-by-example speech search using visual grounding," in *Proc. ICASSP*, 2019.

[40] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[41] D. Harwath and J. R. Glass, "Learning word-like units from joint audio-visual analysis," in *Proc. ACL*, 2017.