Catplayinginthesnow: Impact of Prior Segmentation on a Model of Visually Grounded Speech

William N. Havard[1,2], **Laurent Besacier**[1], **Jean-Pierre Chevrot**[2]

[1] LIG, Univ. Grenoble Alpes, CNRS, Grenoble INP, 38000 Grenoble, France
[2] LIDILEM, Univ. Grenoble Alpes, 38000 Grenoble, France

first-name.lastname@univ-grenoble-alpes.fr

Abstract

The language acquisition literature shows that children do not build their lexicon by segmenting the spoken input into phonemes and then building up words from them, but rather adopt a top-down approach and start by segmenting word-like units and then break them down into smaller units. This suggests that the ideal way of learning a language is by starting from full semantic units. In this paper, we investigate if this is also the case for a neural model of Visually Grounded Speech trained on a speech-image retrieval task. We evaluated how well such a network is able to learn a reliable speech-to-image mapping when provided with phone, syllable, or word boundary information. We present a simple way to introduce such information into an RNN-based model and investigate which type of boundary is the most efficient. We also explore at which level of the network's architecture such information should be introduced so as to maximise its performances. Finally, we show that using multiple boundary types at once in a hierarchical structure, by which low-level segments are used to recompose high-level segments, is beneficial and yields better results than using low-level or high-level segments in isolation.

1 Introduction and Prior Work

Visually Grounded Speech (VGS) models whether CNN-based (Harwath and Glass, 2015; Harwath et al., 2016; Kamper et al., 2017) or RNN-based (Chrupała et al., 2017; Merkx et al., 2019) became recently popular as they enable to model complex interaction between two modalities, namely speech and vision, and can thus be used to model child language acquisition, and more specifically lexical acquisition. Indeed, these models are trained to solve a speech-image retrieval task. That is, given a spoken input description, they are trained to retrieve the image that matches the description the best. This task requires the model to identify lexical units that might be relevant in the spoken input, detect which objects are present in the image, and finally see if those objects match the detected spoken lexical units. Their task is thus very close to that of a child learning its mother tongue, who is surrounded by a visually perceptible context and tries to match parts of the acoustic input to surrounding visible situations. Research in language acquisition have put forward that children do not build their lexicon by segmenting the spoken input into phonemes and then building up words, but rather adopt a top-down approach (Bortfeld et al., 2005) and start by identifying and memorising whole words (Jusczyk and Aslin, 1995) or chunks of words (Bannard and Matthews, 2008) and then segment the spoken input into smaller units, such as phonemes. This suggests that the most efficient way of segmenting the spoken input to map a visual context to its description is at word level. From a more technological point of view, speech-based models lag behind their textual counterparts. For example, speech-image retrieval performs worse than text-image retrieval, despite being trained on the same data, the only changing factor being the modality where text or speech is used as a query. This begs the question: what makes text inherently better than speech for such applications? Is it because text is made up of already-segmented (discrete) units which lack internal variation, or because these discrete units (usually tokens) stand for full semantic units, or a combination of both?

Since the pioneering computational modelling work of lexical acquisition by Roy and Pentland (2002), neural network enabled an even tighter interaction between the visual and the audio modalities. Recent works suggest that networks trained on a speech-image retrieval task perform an implicit segmentation of their input. Whether CNN-based approaches or RNN-based approaches are

Proceedings of the 24th Conference on Computational Natural Language Learning, pages 291–301
Online, November 19-20, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

employed, all seem to segment individual words from the inputted spoken utterance (Harwath et al., 2016; Chrupała et al., 2017; Havard et al., 2019; Havard et al., 2019; Merkx et al., 2019). This result stands also for languages other than English, such as Hindi or Japanese (Harwath et al., 2018; Havard et al., 2019; Azuh et al., 2019; Ohishi et al., 2020). Chrupała et al. (2017) and Merkx et al. (2019), however, observed that not all layers encode word-like units, suggesting that some layers specialise in lexical processing whereas some other do not encode such information.

Contributions Our research question can be framed as follows: what is the segmentation that maximises the performance of an audio-visual network if speech were to be segmented? To answer this question we investigate *how* it is possible to give speech boundary information to a neural network and explore *which* type of boundary (phone, syllable, or word) is the most efficient. We also explore *where* such information should be provided, that is, at which layer of the architecture is the addition of this information the most beneficial?

2 Model & Data

Data We use two different data sets in our experiments: MS COCO (Lin et al., 2014) and Flickr8k (Hodosh et al., 2013). Both corpora were initially conceived for computer vision purposes and both feature a set of images along with five written descriptions of the images. The captions were not computer generated but written by humans. We use the audio extensions of both data sets: for Flickr8k, we use the captions provided by Harwath and Glass (2015), and for COCO we use Synthetic COCO data set introduced by (Chrupała et al., 2017; Chrupała et al., 2017). The captions of Harwath and Glass (2015) were gathered using Amazon Mechanical Turk and were thus uttered by humans. This data set is particularly challenging as it features multiple speakers and the quality of the recording is uneven from one caption to another. The spoken captions of Chrupała et al. (2017) feature synthetic speech generated with Google's Text-to-Speech system. For both corpora, we extracted speech-to-text alignments through the *Maus* forced aligner (Kisler et al., 2017) online platform, resulting in alignments at word and phone levels.

Architecture The models we train in our experiments all have the same architecture and are based on that of Chrupała et al. (2017).[1] As all models of VGS, be they CNN-based or RNN-based, this architecture has two main components: an image encoder and a speech encoder. Such models are trained to solve a speech-image retrieval task, that is, given a query in the form of a spoken description, they should retrieve the closest matching image fitting the description.

The image encoder is a simple linear layer that reduces pre-computed VGG image vectors to the desired dimension. The speech encoder, which receives MFCC vectors as input, consists of a 1D convolutional layer, followed by five stacked recurrent layers with residual connections, followed by an attention mechanism. We use uni-directional recurrent layers and not bi-directional recurrent layers even though it has been shown they lead to better results (Merkx et al., 2019). Indeed, we aim at having a cognitively plausible model: humans process speech in a left-to-right fashion, as speech is being gradually uttered, and not from both ends simultaneously. We use the same loss function as initially used by Chrupała et al. (2017):

$$\mathcal{L}(u,i,\alpha) = \sum_{u,i}\left(\sum_{u'}\max[0,\alpha+d(u,i)-d(u',i)] + \sum_{i'}\max[0,\alpha+d(u,i)-d(u,i')]\right) \tag{1}$$

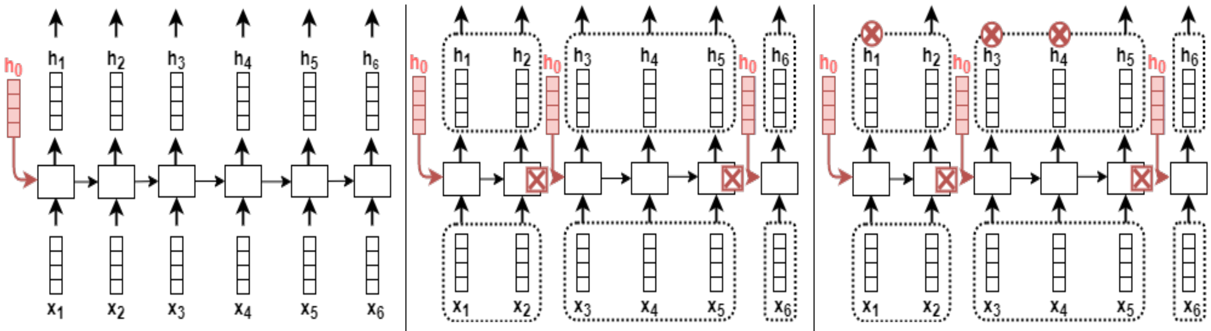
This contrastive loss function encourages the network to minimise the cosine distance d by a margin α between an image i and its corresponding utterance u, while maximising the distance between mismatching image/utterance pairs i'/u and i/u'. In our experiments we set $\alpha = 0.2$.

Hyperparameters For both COCO and Flickr8k we use 1D convolutions with 64 filters of length 6 and a stride of 1 to preserve the original time resolution (and hence, boundary position). We use 512 units per recurrent layer for COCO and 1024 for Flickr8k. All models were trained using Adam optimiser and an initial learning rate of 0.0002. For our experiments we use the pre-computed MFCC vectors and pre-computed VGG vectors provided by Chrupała et al. (2017).[2] We also use the same training, validation and testing splits.[3]

[1] The code we use is based on `https://github.com/gchrupala/vgs`

[2] 12 MFCC coef. plus energy for COCO; 12 MFCC coef. plus energy as well as deltas and delta deltas for Flickr8k.

[3] Training/Validation/Test split contain



(a) (b) (c)

Figure 1: Graphical representation of the different GRUs used in our experiments. Figure 1a shows a Vanilla GRU. Figure 1b shows GRU$_{\text{PACK.}}$ in the ALL condition where all the vectors produced at each time step are passed on to the next layer. 1c shows GRU$_{\text{PACK.}}$ in the KEEP condition where only the last vector of a segment is passed on to the next layer, thus resulting in a output sequence shorter than the input sequence. The red crosses inscribed in a square (☒) signal that the output vector computed at a given timestep is not passed on to the next timestep and that the initial state h_0 is passed on instead. The red crosses inscribed in a circle (⊗) signal that the output vector computed at a given timestep is not passed on to the next layer. Dotted line group vectors belonging to a same segment (either phone, syllable-connected, syllable-word, or word). Note that h_0 is only passed on to the next state at the end of a segment, thus effectively materialising a boundary by manually resetting the history. Also note that the $x_1, x_2, ..., x_t$ figured in this representation could either be the original input sequence (in our case, acoustic vectors) or could also be the output of the previous recurrent layer.

3 Integrating Segmentation Information

3.1 Boundary Types

As previously stated, we are interested in supplying our network with linguistic information such as segment boundaries. We define a segment as either being a phoneme, a syllable, or a word. We consider two different types of syllables. Indeed, when we speak, words are not uttered one after the other in a disconnected fashion, but are rather blended together through a process called "resyllabification". In English, this phenomenon is visible when a word ending with a consonant is followed by a word starting with a vowel. In this case, the final consonant of the first word tends to be detached from it and attached to the next word, thus crossing the word boundary. This phenomenon is illustrated in Example (1) where phonemes in red indicate a resyllabification phenomenon.

(1)	This is an article.
Transcription[4]	/ðɪs#ɪz#ən#ɑɹtɪkəl/
a. *No resyllabification*	/ðɪs.ɪz.ən.ɑɹ.tɪ.kəl/
b. *With resyllabification*	/ðɪ.sɪ.zə.nɑɹ.tɪ.kəl/

For the rest of this article "syllables-word" will refer to syllables that result of a segmentation that does not take into account resyllabification (1-a), whereas "syllables-connected" will refer to syllables that result of a segmentation that takes into account resyllabification (1-b). It should be noted that in the syllables-connected condition, most word boundaries are lost.[5] In the syllables-word condition, however, all word boundaries are preserved and the segmentation inside a word may result in a morphemic segmentation (as for example in "runway" /ɹʌn.weɪ/ or "air.plane" /ɛɹ.plein/). Nevertheless, this is not always the case, especially for longer words that are of non-germanic origin (such as "elephant" /ɛ.lɛ.fant/ or "computer" /kəm.pju.tɚ/). We expect models trained in the syllables-connected condition to perform worse than those trained in the syllables-word condition as resyllabification hinders word recognition (Vroomen and Gelder, 1999).

Segment boundaries were derived from the forced alignment metadata so as to indicate which MFCC vector constitutes a boundary or not.[6] Therefore, for each caption we have a sequence X of length T of d-dimensional acoustic vectors $X = [x_1^d, x_2^d, ..., x_T^d]$ and a corresponding

113,287/5,000/5,000 images (COCO) and 6,000/1,000/1,000 images (Flickr8k).

[4] We use "#" to signal word boundaries and "." to signal syllable boundaries.

[5] Word boundaries are not lost in the following cases: V#V and C#C when CC is not an allowed complex onset. C and V respectively refer to "consonant" and "vowel".

[6] As the force aligner used does not provide alignment at the syllable level, we wrote a custom script to recreate syllables from the phonemic transcription.

sequence of scalars B representing boundaries $B = [b_1, b_2, ..., b_T]$, $b_t \in \{0, 1\}$, where $b_t \triangleq 1$ if x_t is a segment boundary, 0 otherwise.

3.2 Integrating Boundary Information

In order to integrate boundary information into the model, we take advantage of how recurrent neural networks compute their output. They can be formalised as follows:

$$h_t = f\left(h_{t-1}, x_t; \theta\right) \tag{2}$$

where the hidden state at timestep t noted h_t is a function f of the previous hidden state h_{t-1} and the current input at x_t, with θ being learnable parameters of the function f. A special case arises at the very first time step $t = 1$ as h_{t-1} does not exist. In this case, the initial state h_{t-1} noted h_0 is set to be a vector of 0. The output of such a network at timestep T is thus dependent on all the previous timesteps. An illustration of such a network is depicted in Figure 1a. In this work, we use GRUs (Cho et al., 2014), but our methodology is applicable to any other type of recurrent cell such as vanilla RNNs or LSTMs.

Our approach to integrate boundary information into the network can be formalised as follows:

$$h_t = \begin{cases} f\left(h_0, x_t; \theta\right), & \text{if } b_{t-1} = 1 \\ f\left(h_{t-1}, x_t; \theta\right), & \text{otherwise} \end{cases} \tag{3}$$

In our approach, h_t is only dependent on the previous timestep h_{t-1} if the previous timestep was not an acoustic vector corresponding to segment boundary ($b_{t-1} \neq 0$). If the previous timestep corresponds to a segment boundary ($b_{t-1} = 1$), we reset the hidden state so that it is equal to h_0. Hence, vectors in the same segment are temporally dependent, but vectors belonging to two different segments are not. The GRUs that use this computing scheme will from now on be referred to as GRU$_{\text{PACK.}}$, as vectors belonging to the same segment are "packed" together.

We derived two different conditions from this initial setting: ALL and KEEP. In the ALL condition (see Figure 1b), all the vectors belonging to a segment are forwarded to the next layer (which can either be a recurrent layer, or an attention mechanism depending on the position of the GRU$_{\text{PACK.}}$ layer.) In the KEEP condition, only the last vector of each segment is forwarded to the next layer (see Figure 1c). The length of the output and input sequence stays the same in the ALL condition. However, it should be noted that in the KEEP condition, the length of the output sequence is shorter than the input sequence. Potentially, the length of the sequences can be different for different items inside a batch as the captions have a different number of segments (be they phones, syllables or words). For this reason, and as the subsequent layers expect a 3D rectangular matrix,[7] we add padding vectors on the sequence dimension until all the elements of the batch have the same sequence length. The difference between ALL and KEEP is motivated by the fact that we believe that keeping the last vector of a segment could constrain the network to build more consistent representations for different occurrences of the same segment, as the subsequent layers will have less information to rely on. A similar approach to ours was proposed by Chen et al. (2019) in an Audio-Word2Vec experiment, where instead of being given gold segment boundaries, a classifier outputs a probability that a given frame constitutes a segment boundary.

4 Experiments and Results

4.1 GRU$_{\text{PACK.}}$ Position and Random Boundaries

In order to understand where boundary information should be introduced (that is, at which level of the architecture), we train as many models as the number of recurrent layers, where each time one layer of GRUs is replaced with one GRU$_{\text{PACK.}}$ layer. For example, "GRU$_{\text{PACK.}}$–3" refers to a model where the third layer of GRUs is a GRU$_{\text{PACK.}}$ layer and other layers (1^{st},2^{nd},4^{th}, and 5^{th} layer) are vanilla GRU layers. This setting will allow to explore *where* introducing boundary information is the most efficient.

To understand if introducing boundary information helps the network in its task, we compare the performance of the models using boundary information with a baseline model which does not use any (thus, all the recurrent layers of the baseline architecture are Vanilla GRU layers). This model will from now on be referred to as BASELINE. We also introduce another condition, where, instead of training models with real segment boundaries (which from now on will be referred to as TRUE), we train models with random boundaries (which from now on will be referred to as RANDOM). Indeed, it could be that randomly slicing speech into sub-units leads to better results, even though the resulting units do

[7] Of shape batch size × sequence × embedding dimension.

Data set	R@1	R@5	R@10
COCO	9.0	27.0	39.5
Flickr8k	4.3	13.4	21.4

Table 1: Mean recalls at 1, 5, and 10 (in %) on a speech-image retrieval task COCO and Flickr8k in the BASELINE condition. Chance scores are 0.0002/0.001/0.002 for COCO and 0.001/0.005/0.01 for Flickr8k.

not constitute linguistically meaningful units. Consequently, training models with random boundaries will enable us to verify this claim. Random boundaries were generated by simply shuffling the position of the real boundaries (vector B introduced in §3.1), resulting in as many randomly positioned boundaries as there are real ones. Note that we do still expect the models to have reasonable results even when using random boundaries, as acoustic vectors are kept untouched. Nonetheless, we expect that placing random boundaries will hinder network's learning process and thus yield results significantly lower than when using true boundaries. We expect results to be significantly lower in the RANDOM-KEEP condition as this condition is equivalent to randomly subsampling the input, and thus removing a lot of information.

4.2 Evaluation

Models are evaluated in term of Recall@k (R@k). Given a spoken query, R@k evaluates the models ability to rank the target paired image in the top k images. In order to evaluate if the results observed in our different experimental conditions (TRUE-ALL, TRUE-KEEP, RANDOM-ALL, RANDOM-KEEP) are different from one another and from the BASELINE condition, we used a two-sided proportion Z-Test. This test is used to check if there is a statistical difference between two independent proportions. As for each spoken query there is only one target image, R@k becomes a binary value which equals 1 if the target image is ranked in the top k images and 0 otherwise. In our case, the proportion that we test is the number of successes over the number of trials (which corresponds to the number of different caption/image pairs in the test set).

4.3 Results

Overall, our experimental settings led to the training of 81 different models per data set.[8] BASELINE results are shown in Table 1, results for the TRUE/RANDOM conditions obtained on the Flickr8k are shown in Table 2 and results on COCO in Table 5 (Appendix A). We obtain lower results on Flickr8k than on COCO which shows how difficult the task is on natural speech. The results obtained on synthetic speech are also very low compared to their textual counterpart.[9] For space reasons, and as the results on both Flickr8k and COCO show the same trend, we will focus in the following pages on analysing the results obtained on the Flickr8k data set. The results obtained on the COCO data set are reported in Appendix A.

TRUE/RANDOM and ALL/KEEP Boundaries One of the questions our experiments aim at answering is whether introducing boundary information helps the network in solving its task or not. To do so, we first compare the difference between TRUE and RANDOM boundaries. We notice different patterns depending on the position of the $\text{GRU}_{\text{PACK.}}$ layer and also depending on the ALL and KEEP conditions.

We observe that in the ALL condition the results between TRUE and RANDOM boundaries are overall not statistically different from one another, and are not significantly better or worse from the baseline results. There is only one case where such differences are statistically significant: for the 1^{st} layer when using word segments. However, in the KEEP condition, we observe a strong difference between TRUE and RANDOM boundaries across all boundary types and across most of the layers. Overall, in the KEEP condition, models trained with TRUE boundaries have statistically different results from models trained with RANDOM boundaries. Also, in such settings, the results obtained are generally statistically better than the baseline, while in the RANDOM-KEEP condition the results are statistically worse than the baseline.

These results show that there is overall no difference between using TRUE or RANDOM boundaries in the ALL condition (except for one layer), hence showing that boundary information is not used effectively by the network. In contrast, the difference between TRUE and RANDOM in the KEEP condition shows that boundary information is effectively used by the network. Using random boundaries which do not delimit meaningful linguistic units really hurts the performance of the network, espe-

[8] (Seg. type $\in$ {phone,syl.-connected,syl.-word,word} $\times$ $\text{GRU}_{\text{PACK.}}${1,2,3,4,5} $\times$ {TRUE,RANDOM} $\times$ {ALL,KEEP}) + BASELINE

[9] Merkx and Frank (2019) report R@1 $=27.5$ on a GRU-based model using characters as input.

	Flickr8k — KEEP condition								Flickr8k — ALL condition							
GRU	Phones		Syl.-Co.		Syl.-Word		Word		Phones		Syl.-Co.		Syl.-Word		Word	
Pack.	T	R	T	R	T	R	T	R	T	R	T	R	T	R	T	R
5	3.6	**3.7**	***3.6***	*2.5*$^{-}$	**3.3**	3.0	**3.2**	**3.2**	**4.0**	3.9	**4.1**	**4.1**	**4.3**	3.9	3.4	**4.2**
4	**3.8**	**3.8**	**4.4**	3.5	***3.9***	*2.6*$^{-}$	***5.2***$^{+}$	*2.5*$^{-}$	4.0	**4.4**	3.9	**4.1**	**4.3**	3.8	**4.5**	**4.5**
3	***4.9***$^{+}$	*3.8*	***4.5***	*3.1*	***5.3***$^{+}$	*3.1*	***4.9***$^{+}$	*3.3*	**4.5**	4.4	**4.3**	4.2	**4.4**	4.2	**4.5**	3.8
2	**4.8**$^{+}$	3.9	***5.1***$^{+}$	*3.6*	***4.8***	*3.4*	***5.4***$^{+}$	*3.4*	**4.5**	3.8	***4.8***	*3.6*	**4.4**	4.2	**4.7**	4.1
1	***4.8***	*2.4*$^{-}$	***3.4***	*1.9*$^{-}$	***4.4***	*2.0*$^{-}$	***3.9***	*1.9*$^{-}$	**4.3**	3.4	**4.0**	**4.0**	**4.4**	4.3	***5.3***$^{+}$	*4.1*

Table 2: Maximum R@1 (in %) for each model trained on test set of the Flickr8k data set (models were selected based on the maximum R@1 on the validation set). "T" stands for TRUE (boundaries) and "R" stands for RANDOM (boundaries). "Syl-Co." and "Syl-Word" stand for "Syllables-Connected" and "Syllables-Word" respectively. Each line shows the results for when a specific recurrent layer is a GRU$_{\text{PACK.}}$ layer (see §4.1). The 1^{st} layer is the lowest layer and the 5^{th} the highest. The highest R@1 in the table is shown in red. Best results between each TRUE and RANDOM pair (columnwise) are shown in **bold**. ◌$^{+}$ and ◌$^{-}$ indicate that the results are statistically better (respectively worse) than the baseline. Results in *italics* show statistical significance (two-sided Z-Test, p-value $< 1e^{-2}$, see §4.2) between each TRUE and RANDOM pair (columnwise).

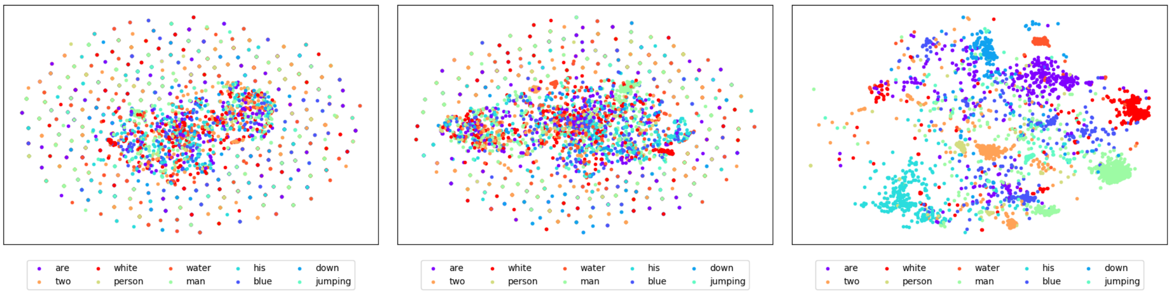
cially in the KEEP condition as most of the vectors are removed. In the ALL condition, using TRUE or RANDOM boundaries yields results close to that of the BASELINE, suggesting boundary information might act as noise and help the network regularise. Thus, as expected, the network was effectively constrained to learn better representations in the KEEP condition. We believe it is the case because in the ALL condition, boundary information is diluted among the neighbouring vectors while this is not the case in the KEEP condition, as each segment is represented by a single vector.

Phones, Syllables, or Words From now on, we will focus on the results obtained in the KEEP condition, as the ALL condition brings only slight improvement over the BASELINE condition. In our experiments we used four different type of segments corresponding to different type of linguistic units: phones, syllables-connected, syllables-word, and words. These different type of segments vary in *length* (words and syllables are longer than phones), *quantity* (there are more phones and syllables than words), and *intrinsic linguistic information*: phones only show which are the basic acoustic units of the language, while word segments represent meaningful units, and syllables-word and syllables-connected are a higher form of acoustic unit that may contain morphemic information. Given the task the network is trained for (speech-image retrieval), we do not expect these different units to perform equally well. Indeed, as this task implies mapping an image vector describing which objects are present in a picture and a spoken description of an image, we expect word-like segments (or segments that preserve word boundaries and that bear a substantial amount of semantic information) to perform better.

This is in fact what we observe in practice: word units obtain statistically better results ($R@1 = 5.4$) than the baseline (+1.1pp). Syllables-word also bring significant improvement ($R@1 = 5.3$), however, slightly less than when using word units. It should be noted that syllables-connected segments also obtain statistically significant improvement over the baseline (GRU$_{\text{PACK.}}$–2) despite not preserving all the word boundaries. However, these results are slightly worse than the syllables-word and word segments, suggesting that preserving word boundaries is a property that helps the network. It appears that the size of a segment is also an important parameter. Indeed, phone segments (naturally) preserve word boundaries, but of course naturally lack the internal cohesion of a morpheme or a word as nothing links two adjacent phonemes together. Thus, it seems that segments that preserve meaning (such as words) or from which meaning can be more easily recomposed (syllable) may facilitate the network's task. The fact that syllable-like segments perform as well as word segments might only be an artefact of using English where a high proportion of word is monosyllabic.[10] Working on a language such as Japanese where the syllable-to-morpheme ratio is higher would be a future line of work that would enable to test this hypothesis.

GRU$_{\text{PACK.}}$ Layer Position We introduced boundary information at different levels of our architecture in order to better understand at which layer it is the most useful to add such information.

[10] Jespersen (1929) estimates that at least 8,000 commonly frequent words are monosyllabic in English.



(a) (b) (c)

Figure 2: t-SNE projections of the final vector of different occurrences of eigth randomly selected words (Flick8k) in the BASELINE condition (2a), in the ALL condition (2b), and in the KEEP condition (2c). Plot 2a shows that the representation learnt in the BASELINE and ALL conditions are not word-based as the final vectors of different occurrences of the same word do not cluster together. In the KEEP condition, the model succeeded in learning similar representations for different occurrences of the same word as words cluster together.

Our results clearly show that introducing boundary information at different layers has a clear impact on the results: using such information at the first or the fifth layer is useless, as we notice it either yields similar results to the baseline or worsens the results regardless of the type of boundary used ($\text{GRU}_{\text{PACK.}}$–5). When using syllable-word segments the best results are obtained when introducing the information at $\text{GRU}_{\text{PACK.}}$–3, and at $\text{GRU}_{\text{PACK.}}$–2 when using word segments. Word-like segments seem to be the most robust representation to be used as they yield significantly better results at three different layers ($\text{GRU}_{\text{PACK.}}$–2,3,4). We also notice that phone segments bring no improvement over the baseline at $\text{GRU}_{\text{PACK.}}$–4,5 showing that these layers do not handle phone like information. All in all, these results are in line with that of Chrupała et al. (2017) who found that the intermediate representations of the fifth layer is the less informative in predicting word presence, while lower layers encode this information better. This confirms that the middle layers of our architecture deal with lexical units whereas the fifth layer encodes information that disregards that type of information.

4.4 Segmentation as a Means for Compression

Recall that in the KEEP condition, only the last vector comprising a segment is kept while the other vectors are discarded. This can be interpreted as a form of "guided" subsampling, as usually subsampling does not take into consideration linguistic factors. To understand how much information is kept between the input and the output of a $\text{GRU}_{\text{PACK.}}$ layer in the KEEP condition, we compute an average compression rate (in %) for each of the segment types for Flickr8k. The results are the following: phones = 90.57%, syllables-connected = 93.41%, syllables-word = 94.36%, and words = 94.90%.[11] When we re-analyse our results in light of this information, it appears we can remove a large part of the original input (up to 94.90% if using word segments) while conserving or increasing the original R@1. It is not simply the effect of subsampling that helps, but subsampling with *meaningful* linguistic units. The effect of informed subsampling is striking when we compare R@1 for RANDOM-KEEP, which are always below the BASELINE, while TRUE-KEEP are on a par with the BASELINE or better. A counter-intuitive finding of our experiments is that it is better to subsample early on (in the first layers) and thus remove most of the information early on than later on. Subsampling with word segments in $\text{GRU}_{\text{PACK.}}$–2 (and thus only keeping 5.1% of the original amount of information for the subsequent layers) yields better results than subsampling with the same resolution at $\text{GRU}_{\text{PACK.}}$–5.

5 Towards Hierarchical Segmentation

In our current approach, only one out of the five recurrent layers is a $\text{GRU}_{\text{PACK.}}$ layer, which handles only one type of segment. However, we can stack as many $\text{GRU}_{\text{PACK.}}$ as desired, provided they are supplied with boundary information. Stacking such layers enables us to not only integrate bound-

[11] Note that the compression rate for syllables-word and words is very close, suggesting there is a significant overlap between syllables-word units and word units.

Architecture	**5 layers**					**4 layers**				**3 layers**			**2 layers**	
2^{nd}GRU$_{\text{PACK.}}$ / 1^{st}GRU$_{\text{PACK.}}$	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**1**	**2**
1		7.7	7.7	7.3	3.9		7.6	7.9	5.7		**8.1**	5.3		**6.4**
2			**8.2**	7.6	5.8			**8.1**	6.3			7.3		
3				7.1	6.5				6.7					
4					6.1									
5														
Baseline (No GRU$_{\text{PACK.}}$)	4.3					4.4				3.4			3.5	

Table 3: R@1 obtained on the test set of the Flickr8k data set with a hierarchical architecture consisting of two GRU$_{\text{PACK.}}$ layers using phone and word segments (models were selected based on the maximum R@1 on the validation set). Best score overall is shown in red. Best score (layer-wise) is shown in **bold**. Greyed out cell signal impossible configurations. We also indicate R@1 obtained on a baseline architecture without any GRU$_{\text{PACK.}}$.

ary information, but also introduce structure, where one layer handles one type of segment (e.g. phone) and the following GRU$_{\text{PACK.}}$ layer handles another type of segment, that is hierarchically above the preceding (e.g. syllable, or word).[12] Harwath et al. (2020) explored such hierarchical architecture using a CNN-based model that incorporated vector quantisation layers and found that it improves R@k. Our work thus attempts to verify if it is also the case for an RNN-based model.

Phones and Words: We first explore hierarchical segmentation with phones and words on the Flickr8k data set.[13] We only consider the KEEP condition as it yields better results than the ALL condition. We vary the position of the GRU$_{\text{PACK.}}$ layers as well as the number of layers (from 2 to 5) and test all possible positions with two GRU$_{\text{PACK.}}$ layers. For each configuration, the lowest GRU$_{\text{PACK.}}$ will receive phone boundary information, and the next GRU$_{\text{PACK.}}$ layer will receive word boundary information. Note that such configuration results in a double sequence reduction. Indeed, after the first GRU$_{\text{PACK.}}$ layer, they are only as many output vectors as there are phones, and in the second, the resulting phone vectors are recomposed together to form words, resulting in as many output vectors as there are words. Results are shown in Table 3. Training an architecture with two GRU$_{\text{PACK.}}$ layers, each handling two different types of segments results in much better R@1 than the baseline (+3.9pp) and than a single-GRU$_{\text{PACK.}}$-layered architecture (+2.8pp), thus showing that introducing hierarchy is beneficial. Results also confirm that the layer 2 and 3 of our architecture are those that benefit the most from adding linguistic information, and confirm the fact that the upper layers (such as the fifth) do not take as much advantage of this information as the lower layers. Introducing structure allowed us to remove two recurrent layers without a big loss of performance ($R@1 = 8.1$ for a three-layered architecture with two GRU$_{\text{PACK.}}$ layers) while the baseline architecture with only three layers performs poorly.

Phones, Syllables, and Words: We also explore an architecture with 3 GRU$_{\text{PACK.}}$ layers, to which we provide phone, syllable-word and word boundaries. As in our previous experiments, we vary the number of layers (from 3 to 5), and test all possible configurations. The results of this experiment are presented in Table 4. We notice that the best result obtained with this architecture is far superior to the best result of a single-layered architecture ($R@1 = 9.6, +4.2$pp), but also superior to the best result of a double-layered architecture (+1.4pp over the phone-word architecture). Our best results are obtained by a five-layered architecture with GRU$_{\text{PACK.}}$ in position 1, 3 and 4. However, we notice that the four-layered architecture obtains more consistent results across all layers, the maximum result being only -0.3pp away from best five-layered architecture. We also notice that the 3 layered architecture obtains a very high R@1 of 8.0 which is about two times over the baseline results.

Our results show that the more structure we introduce into the network, the better it performs. Additionally, introducing structure enables us to reduce the number of layers (and hence the number of computations) while increasing the performances compared to the baseline. Overall, it is better to

[12] Note that it could also be possible to use larger units, such as chunks.

[13] We also explored two other hierarchical architectures that use phones and syllables-word, and syllables-word and words. The results are reported in Appendix B in Table 6 and Table 7.

use boundary information in coordination in a hierarchical structure than using them in isolation.

Architecture GRU$_{\text{PACK.}}$	**5 layers**	**4 layers**	**3 layers**
1 + 2 + 3	8.5	**9.3**	8.0
1 + 2 + 4	8.1	8.6	
1 + 2 + 5	7.8		
1 + 3 + 4	**9.6**	8.4	
1 + 3 + 5	7.9		
1 + 4 + 5	7.8		
2 + 3 + 4	8.8	8.3	
2 + 3 + 5	8.5		
2 + 4 + 5	8.3		
3 + 4 + 5	7.8		

Table 4: R@1 obtained on the test set of the Flickr8k data set with a hierarchical architecture consisting of three GRU$_{\text{PACK.}}$ layers using phone, syllable-word and word segments (models were selected based on the maximum R@1 on the validation set).

6 Discussion and Future Work

The goal of our experiments is to see if segmenting speech in sub-units is beneficial, and if so, which units maximise the performance. It is indeed the case that segmenting speech into sub-units helps. As to which segment obtains the best performance we observe mixed results. Indeed, word segmentation yields better results than phone segmentation, but we do also observe that syllable-like segmentation also gives results that are in the same ballpark as word segmentation. Nevertheless, word segmentation seems to be a *more robust* representation compared to syllable as such word segments consistently yield better results at various levels of our architecture.

Another finding of our experiments which we believe is important is that one cannot subsample speech without taking into account its linguistic nature. Indeed, random subsampling might yield results on a par with the baseline, but this might only be a regularisation effect. *Linguistically informed subsampling* (KEEP condition) yields however much better results and should be favoured.

Regarding the question of why textual approaches perform better than spoken approaches, we conclude that the fact that tokens stand for full semantic units plays little in their performance. The fact that text-based models use segmented input (either tokens or characters) also seems to play little in the final performance, otherwise we should have observed better results as our input was also segmented. What seems most crucial is that the representation of a token never changes whereas speech exhibits lots of variation, as no word is pronounced exactly in the same fashion when uttered. Our approach helped the network in building more consistent representations for the same word (especially in the KEEP condition, see Figure 2), even though it did not succeed for every word. Consistent representation across various occurrences seems to be the most important factor.

Finally, our experiments allowed us to observe that, such as for humans, the use of large units, such as words, is indeed the most efficient solution to learn a reliable speech-to-image mapping. Indeed, even if phone and syllable-like units yield non trivial results, they are less robust than word segments. Our GRU$_{\text{PACK.}}$ setting also allowed us to simply introduce hierarchy in a neural network by simply stacking GRU$_{\text{PACK.}}$ layers and providing different boundary information to each of them. Our experiments allowed us to confirm the results obtained by Harwath et al. (2020) on a CNN-based VGS model, stating that introducing hierarchical structures proves beneficial overall even for shallower architectures. Interestingly, our hierarchical experiments allowed us to notice that using segment boundaries in isolation only brings slight improvements. It is only when different levels are combined (phones and words, or phones, syllables and words) that the performance of the network reaches its peak.

The future lines of work we imagine consist in *learning* where the boundaries are located instead of supplying boundary information to the network at training and testing time. We could indeed use ACT recurrent cells (Kreutzer and Sokolov, 2018) or an architecture such as (Chen et al., 2019) that would dynamically and unsupervisedly learn how to segment the input signal into sub-units. The additional advantage of such methods is that they make no presupposition on the form/size of the segments, and consequently on what a good segment should or should not be, but lets the network find the optimal solution. Finally, we plan to also introduce syntactic information and integrate chunk boundaries and measure the impact of syntactical grouping of spoken units.

Acknowledgments

This work was supported by grants from NeuroCoG IDEX UGA as part of the “Investissements d’avenir” program (ANR-15-IDEX-02).

References

Emmanuel Azuh, David Harwath, and James Glass. 2019. Towards Bilingual Lexicon Discovery From Visually Grounded Speech Audio. In *Proc. Interspeech 2019*, pages 276–280.

Colin Bannard and Danielle Matthews. 2008. Stored word sequences in language learning: The effect of familiarity on children's repetition of four-word combinations. *Psychological Science*, 19(3):241–248. PMID: 18315796.

Heather Bortfeld, James L. Morgan, Roberta Michnick Golinkoff, and Karen Rathbun. 2005. Mommy and me: Familiar names help launch babies into speech-stream segmentation. *Psychological Science*, 16(4):298–304. PMID: 15828977.

Y. Chen, S. Huang, H. Lee, Y. Wang, and C. Shen. 2019. Audio word2vec: Sequence-to-sequence autoencoding for unsupervised learning of audio segmentation and representation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(9):1481–1493.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734. Association for Computational Linguistics.

Grzegorz Chrupała, Lieke Gelderloos, and Afra Alishahi. 2017. Representations of language in a model of visually grounded speech signal. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 613–622. Association for Computational Linguistics.

Grzegorz Chrupała, Lieke Gelderloos, and Afra Alishahi. 2017. Synthetically spoken coco. [Data set] `http://doi.org/10.5281/zenodo.400926`.

D. Harwath, G. Chuang, and J. Glass. 2018. Vision as an interlingua: Learning multilingual semantic embeddings of untranscribed speech. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4969–4973.

David Harwath and James Glass. 2015. Deep multimodal semantic embeddings for speech and images. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 237–244.

David Harwath, Wei-Ning Hsu, and James R. Glass. 2020. Learning hierarchical discrete linguistic units from visually-grounded speech. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

David Harwath, Antonio Torralba, and James R. Glass. 2016. Unsupervised learning of spoken language with visual context. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 1866–1874, Red Hook, NY, USA. Curran Associates Inc.

William N. Havard, Jean-Pierre Chevrot, and Laurent Besacier. 2019. Models of Visually Grounded Speech Signal Pay Attention to Nouns: A Bilingual Experiment on English and Japanese. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8618–8622.

William N. Havard, Jean-Pierre Chevrot, and Laurent Besacier. 2019. Word Recognition, Competition, and Activation in a Model of Visually Grounded Speech. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 339–348, Hong Kong, China. Association for Computational Linguistics.

Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899.

Otto Jespersen. 1929. *Monosyllabism in English*.

P.W. Jusczyk and R.N. Aslin. 1995. Infants′ detection of the sound patterns of words in fluent speech. *Cognitive Psychology*, 29(1):1–23.

Herman Kamper, Shane Settle, Gregory Shakhnarovich, and Karen Livescu. 2017. Visually grounded learning of keyword prediction from untranscribed speech. pages 3677–3681.

Thomas Kisler, Uwe Reichel, and Florian Schiel. 2017. Multilingual processing of speech via web services. *Computer Speech & Language*, 45:326 – 347.

Julia Kreutzer and Artem Sokolov. 2018. Learning to segment inputs for NMT favors character-level processing. *Proceedings of the International Workshop on Spoken Language Translation October 29-30, 2018 Bruges, Belgium*, 1:166–172.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, C. Lawrence Zitnick, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars. 2014. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.

Danny Merkx and Stefan L. Frank. 2019. Learning semantic sentence representations from visually grounded language without lexical knowledge. *Natural Language Engineering*, 25(4):451–466.

Danny Merkx, Stefan L. Frank, and Mirjam Ernestus. 2019. Language Learning Using Speech to Image Retrieval. In *Proc. Interspeech 2019*, pages 1841–1845.

Y. Ohishi, A. Kimura, T. Kawanishi, K. Kashino, D. Harwath, and J. Glass. 2020. Trilingual semantic embeddings of visually grounded speech with self-attention mechanisms. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4352–4356.

Deb K. Roy and Alex P. Pentland. 2002. Learning words from sights and sounds: a computational model. *Cognitive Science*, 26(1):113–146.

Jean Vroomen and Beatrice De Gelder. 1999. Lexical access of resyllabified words: Evidence from phoneme monitoring. *Memory & Cognition*, 27(3):413–421.