Textual supervision for visually grounded spoken language understanding

Bertrand Higy
Cognitive Science and AI
Tilburg University
b.j.r.higy@uvt.nl

Desmond Elliott
Laboratory for Multimodal Processing
University of Copenhagen
de@di.ku.dk

Grzegorz Chrupała
Cognitive Science and AI
Tilburg University
g.chrupala@uvt.nl

Abstract

Visually-grounded models of spoken language understanding extract semantic information directly from speech, without relying on transcriptions. This is extremely useful for low-resource languages, where transcriptions can be expensive or impossible to obtain. Recent work showed that these models can be improved if transcriptions are available at training time. However, it is not clear how an end-to-end approach compares to a traditional pipeline-based approach when one has access to transcriptions. Comparing different strategies, we find that the pipeline approach works better when enough text is available. With low-resource languages in mind, we also show that translations can be effectively used in place of transcriptions but more data is needed to obtain similar results.

1 Introduction

Spoken language understanding promises to revolutionize how we interact with technology by allowing people to interact with electronic devices through voice commands. However, mapping speech to meaning is far from trivial. The traditional approach, which has proven its effectiveness, relies on text as an intermediate representation. This so-called pipeline approach combines an automatic speech recognition (ASR) system and a natural language understanding (NLU) component. While this allows us to take advantage of improvements achieved in both fields, it requires transcribed speech, which is an expensive resource.

Visually-grounded models of spoken language understanding (Harwath et al., 2016; Chrupała et al., 2017) were recently introduced to extract semantic information from speech directly, without relying on textual information (see Figure 1 for an illustration). The advantages of these approaches are twofold: (i) expensive transcriptions are not necessary to train the system (which is of particular interest for low-resource languages) and (ii) trained in an end-to-end fashion, the whole system is optimized for the final task, which in other applications has been shown to give better performance. While text is not necessary to train such systems, recent work has shown that they can greatly benefit from textual supervision if available (Chrupała, 2019; Pasad et al., 2019), generally using the multitask learning setup (MTL) (Caruana, 1997).

Figure 1: An image described by an English spoken caption (represented by its spectrogram), its transcription, and translation into Japanese. Visually-grounded models are usually trained to map the image and its spoken caption into a shared semantic space.

However, the end-to-end MTL-based models in previous works have not been compared against the more traditional pipeline approach that uses ASR as an intermediate step. The pipeline approach could be a strong baseline as, intuitively, written transcriptions are an accurate and concise represen-

tation of spoken language. In this paper, we set out to determine the relative differences in performance between end-to-end approaches and pipeline-based approaches. This study provides insights from a pragmatic point of view, as well as having important consequences for making further progress in understanding spoken language.

We also explore the question of the exact nature of the textual representations to be used in the visually-grounded spoken language scenario. The text used in previous work were *transcriptions*, which are a relatively faithful representation of the form of the spoken utterance. Other possibilities include, for example, subtitles, which tend to be less literal and abbreviated, or translations, which express the meaning of the utterance in another language. We focus on the case of translations due to the relevance of this condition for low-resource languages: some languages without a standardized writing system have many millions of speakers. One example is Hokkien, spoken in Taiwan and Southeast China: it may be more practical to collect translations of Hokkien into Mandarin than to get them transcribed. The question is whether translations would be effective as a source of textual supervision in visually-grounded learning.

In summary, our contributions are the following.

- We compare different strategies for leveraging textual supervision in the context of visually-grounded models of spoken language understanding: we compare a pipeline approach, where speech is first converted to text, with two end-to-end MTL systems. We find that the pipeline approach tends to be more effective when enough textual data is available.

- We analyze how the amount of transcribed data affects performance, showing that end-to-end training is competitive only in very limited text conditions; however, textual supervision via transcribed data is marginally effective at this stage.

- We explore the possibility of replacing transcriptions with written translations. In the case of translations, an end-to-end MTL approach outperforms the pipeline baselines; we also observe that more data is necessary with translations than with transcriptions, due to the more complex task faced by the system.

2 Related work

2.1 Visually-grounded models of spoken language understanding

Recent work has shown that semantic information can be extracted from speech in a weakly supervised manner when matched visual information is available. This approach is usually referred to as visually-grounded spoken language understanding. While original work focused on single words (Synnaeve et al., 2014; Harwath and Glass, 2015), the concept has quickly been extended to process full sentences (Harwath et al., 2016; Chrupała et al., 2017). Applied on datasets of images with spoken captions, this type of model is typically trained to perform a speech-image retrieval task where an utterance can be used to retrieve an image which it is a description of (or vice-versa). This is achieved through a triplet loss between images and utterances (in both directions).

A similar approach is used by Kamper and Roth (2018) to perform semantic keyword spotting. They include an image tagger in their model to provide tags for each image in order to retrieve sentences that match a keyword semantically, i.e. not only exact matches but also semantically related ones.

2.2 Textual supervision

A common thread in all of these studies is that the spoken sentences do not need to be transcribed, which is useful due to the cost attached to textual labeling. Subsequent work, however, showed that textual supervision, if available, can substantially improve performance. Chrupała (2019) uses transcriptions through multitask learning. The find that adding a speech-text matching task, where spoken captions have to be match with corresponding transcriptions, is particulary helpful. Pasad et al. (2019) applied the same idea to semantic keyword spotting with similar results. They also examine the effect of decreasing the size of the dataset.

Hsu et al. (2019) explore the use of visually grounded models to improve ASR through transfer learning from the semantic matching task; in contrast, we are interested in improving the performance of the grounded model itself using textual supervision.

2.3 Alternative architectures

Another area of related work is found in the spoken command understanding literature. Haghani et al. (2018) compare different architectures making use

of textual supervision, covering both pipeline and end-to-end approaches. The models they explore include an ASR-based multitask system similar to the present work. For the pipeline system, they try both independent and joint training of the ASR and NLU components. Their conclusion is that an intermediate textual representation is important to achieve good performance and that jointly optimizing the different components improves predictions. Lugosch et al. (2019) propose a pretraining method for end-to-end spoken command recognition that relies on the availability of transcribed data. However, while this pretraining strategy brings improvement over a system trained without transcripts, the absence of any other text-based baseline (such as a pipeline system) prevents any conclusion on the advantage of the end-to-end training when textual supervision is available.

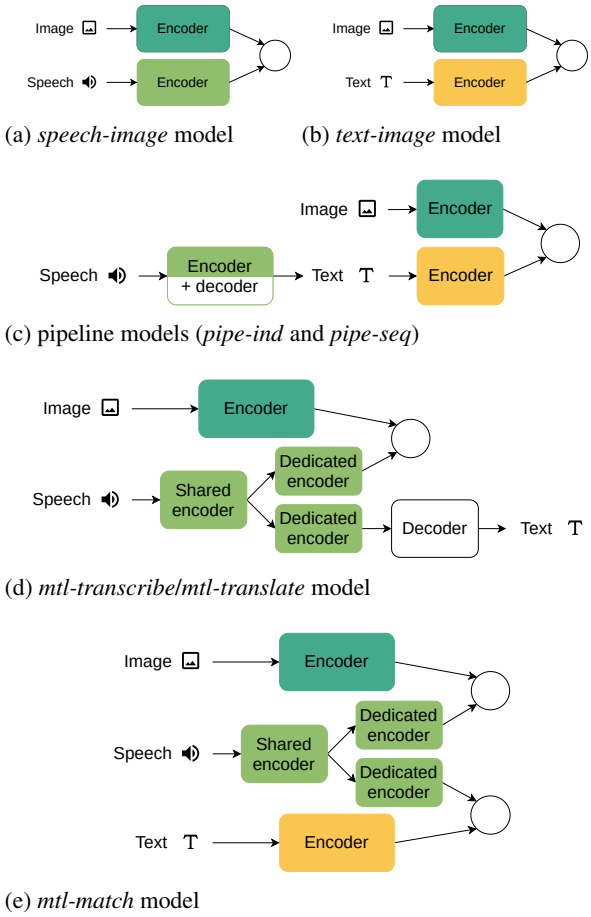
2.4 Multilingual data

While the idea of using multilingual data is not new in the literature, existing work focused on using the same modality for the two languages, either text or speech. Gella et al. (2017) and Kádár et al. (2018) showed that textual descriptions of images in different languages can be used in conjunction to improve the performance of a visually-grounded model, while Harwath et al. (2018) focuses on speech, exploring how spoken captions in two different languages can be used simultaneously to improve performance.

In contrast, our multilingual experiments focus on the setting where speech data from a low-resource language is used in conjuction with corresponding translated written captions. Directly mapping speech to textual translation, or spoken language translation (SLT), has received increasing intereset lately. Following recent trends in ASR and machine translation, end-to-end approaches in particular have drawn much attention, showing competitive results agains pipeline systems (Bérard et al., 2016; Weiss et al., 2017).

3 Methodology

The architecture and the training procedure used in this paper are inspired by the improved version of the visually-grounded spoken language understanding system presented in Merkx et al. (2019). Appendix A.1 provides more details on the choice of hyperparameters.



(a) *speech-image* model (b) *text-image* model

(c) pipeline models (*pipe-ind* and *pipe-seq*)

(d) *mtl-transcribe/mtl-translate* model

(e) *mtl-match* model

Figure 2: Architecture of the different models.

3.1 Architectures

We will compare six different models, based on five architectures (summarized in Figure 2).

Two models serve as reference, the original speech-image matching system (Harwath et al., 2016; Chrupała et al., 2017) that does not rely on text and a text-image model that works directly on text (and thus requires text even at test time; similar to the *Text-RHN* model from Chrupała et al. (2017) or the *Char-GRU* model of Merkx et al. (2019)).

We then have the four core models which can leverage text during training but can also work with speech only at test time. Those are the four models we are mainly interested in. They comprise: two pipeline models, which only differ in their training procedure, and two multitask systems, using either a speech-text retrieval task (similar to what is done in Chrupała (2019) and Kamper and Roth (2018)) or ASR as the secondary target.

3.1.1 The speech-image baseline

The speech-image baseline (*speech-image*) is composed of two main components: an image encoder

and a speech encoder (see Figure 2a).

Image encoder. The image encoder is composed of a single linear layer projecting the image features (see Section 3.4.2) into the shared semantic embedding space (dimension 2048), followed by a normalization layer (ℓ^2 norm).

Speech encoder. The speech encoder is applied on the mel-frequency cepstrum coefficient (MFCC) features described in Section 3.4.2 and composed of a 1D convolutional layer (kernel size 6, stride 2 and 64 output channels), followed by bidirectional gated recurrent units (GRUs) Cho et al. (2014) (4 layers, hidden state of dimension 1024). A vectorial attention layer is then used to convert the variable length input sequence to a fixed-size vector of dimension 2048. Finally, a normalization layer (ℓ^2 norm) is applied.

3.1.2 The text-image baseline

The text-image baseline (*text-image*) measures the possible performance if text is available at test time. It serves as a high estimate of what the four core models could achieve. Those models could theoretically perform better than the *text-image* baseline by taking advantage of information available in speech and not in text. However, extracting the equivalent of the textual representation from speech is not trivial so we expect them to perform worse.

The *text-image* model is comprised of an image encoder and a text encoder (see Figure 2b). The image encoder is identical to the *speech-image* model.

Text encoder. The text encoder is character-based and maps the input characters to a 128-dimensional space through an embedding layer. The output is then fed to a bidirectional GRU (2 layers, hidden state of dimension 1024). A vectorial attention mechanism followed by a normalization layer (ℓ^2 norm) summarizes the variable-length sequence into fixed-length vector (dimension 2048).

3.1.3 The pipeline models

We trained two pipeline models (*pipe-ind* and *pipe-seq*) which only differ in their training procedure (see Section 3.2.2). The architecture (summarized in Figure 2c) is basically composed of an ASR module which maps speech to text, followed by the *text-image* system we just described (our NLU component). The same architecture is used when training with Japanese captions, though the first part is then referred to as the SLT module.

ASR/SLT module. The ASR/SLT module is an attention-based encoder-decoder system which can itself be decomposed into two sub-modules, an encoder and an attention-based decoder. The encoder is similar to the speech encoder described above: it is composed of the same convolutional layer followed by a bidirectional GRU (5 layers, hidden state of dimension 768) but lacks the attention and normalization layers. The attention-based decoder uses a timestep-dependent attention mechanism (Bahdanau et al., 2015) to summarize the encoded input sequence into fixed-size context vectors (one per output token). The recurrent decoder generates the output sequence one character at a time. At each time step, it takes the current context vector and the previous character as input. It is composed of a unidirectional GRU (1 layer, hidden state of dimension 768), a linear projection and the softmax activation layer.

3.1.4 The ASR/SLT-based multitask model

The ASR/SLT-based multitask model (*mtl-transcribe* and *mtl-translate* respectively) combines the *speech-image* model with an ASR/SLT system (similar to the one used in the pipeline models). To do so, the speech encoder of the *speech-image* system and the encoder of the ASR/SLT sytem are merged in a single speech encoder composed of a shared network followed by two task-specific networks (see Figure 2d). The image encoder and the attention-based decoder being identical to the ones described previously, we will focus on the partially-shared speech encoder. The multitask training procedure will be described in Section 3.2.3.

The *mtl-transcribe*/*mtl-translate* speech encoder. The shared part of the speech encoder is composed of a convolutional layer (same configuration as before) and a bidirectional GRU (4 layers, hidden state of dimension 768). The part dedicated to the secondary ASR/SLT task is only composed of an additional bidirectional GRU (1 layer, hidden state of dimension 768). The part dedicated to the speech-image retrieval task is composed of the same GRU layer but also incorporates the vectorial attention and normalization layers necessary to map the data into the audio-visual semantic space.

3.1.5 The text-as-input multitask model

The other multitask model (*mtl-match*) is based on Chrupała (2019). It combines the *speech-image* baseline with a speech-text retrieval task (see Figure 2e). Images and text are encoded by subnetworks identical to the image encoder described in

Section 3.1.1 and the text encoder described in Section 3.1.2 respectively.

The *mtl-match* speech encoder. Similarly to the *mtl-transcribe/mtl-translate* architecture, the speech encoder is composed of a shared component and two dedicated parts. The shared encoder is again composed of a convolutional layer (same configuration as before), followed by a bidirectional GRU (2 layers, hidden state of dimension 1024). The part of the encoder dedicated to the *speech-image* task is composed of a GRU (2 layers, hidden state of dimension 1024), followed by the vectorial attention mechanism and the normalization layer (ℓ^2 norm). Its counterpart for the speech-text task is only made of the vectorial attention mechanism and the ℓ^2 normalization layer.

3.2 Training procedure

3.2.1 Losses

Retrieval loss. The main objective used to train our models is the triplet loss used by Harwath and Glass (2015). The goal is to map images and captions in a shared embedding space where matched images and captions are close to one another and mismatched elements are further apart. This is achieved through optimization of following loss:

$$\sum_{\substack{(u,i)\\(u',i')\neq(u,i)}} \Big(\max(0, d(u,i) - d(u',i) + \alpha) + \max(0, d(u,i) - d(u,i') + \alpha) \Big), \quad (1)$$

where (u, i) and (u', i') are each a pair of matching utterance and image from the current batch, $d(\cdot, \cdot)$ is the cosine distance between encoded utterance and image, and α is some margin (a value of 0.2 is used in practice).

Similarly, a network can be trained to match spoken captions with corresponding transcriptions/translations, replacing utterance and image pairs (u, i) with utterance and text pairs (u, t).

ASR/SLT loss. The ASR and SLT tasks are optimized through the usual cross-entropy loss between the decoded sequence (using greedy decoding) and the ground truth text.

3.2.2 Training the pipeline systems

We use two strategies to train the pipeline systems:

- The **independent** training procedure (*pipe-ind* model), where each module (ASR/SLT and NLU) is trained independently from the other. Here the text encoder is trained on ground-truth written captions or translations.

- The **sequential** training procedure (*pipe-seq* model), where we first train the ASR/SLT module. Once done, we decode each spoken caption (with a beam search of width 10), and use the output to train the NLU system. Doing so reduces the mismatch between training and testing conditions, which can affect the performance of the NLU component. This second procedure is thus expected to perform better than the independent training strategy.

3.2.3 Multitask learning

The *mtl-transcribe/mtl-translate* and *mtl-match* strategies make use of multitask learning through shared weights. To train the models, we simply alternate between the two tasks, updating the parameters of each task in turn.

3.2.4 Optimization procedure

The optimization procedure is inspired by Merkx et al. (2019). We use Adam optimizer (Kingma and Ba, 2015) with a cyclic learning rate (Smith, 2017) varying from 10^{-6} and 2×10^{-4}. All networks are trained for 32 epochs, and unlike Merkx et al., we do not use ensembling.

3.3 Evaluation metrics

Typical metrics for retrieval tasks are recall at n (R@n with n $\in \{1, 5, 10\}$) or median rank (Medr). To compute these metrics, images and utterances are compared based on the cosine distance between their embeddings and a ranked list of images is computed for each utterance (in order of increasing distance). One can then compute the proportion of utterances for which the paired image appears in the top n images (R@n), or the median rank of the paired image over all utterances. For brevity, we only report results with R@10 in the core of the paper. The complete set of results is available in Appendix A.2.

For ASR and SLT, we report word error rate (WER) and BLEU score respectively, using beam decoding in both cases (with a beam width of 10).

All results we report are the mean over three runs of the same experiment with different random seeds.

3.4 Experimental setup

3.4.1 Datasets

The visually grounded models presented in this paper require pairs of images and spoken captions for training. For our experiments on textual supervision, we additionally need the transcriptions corresponding to those spoken captions, or alternatively a translated version of these transcripts. We obtain these elements from a set of related datasets:

- Flickr8K (Hodosh et al., 2013) offers 8,000 images of everyday situations gathered from the website flickr.com together with English written captions (5 per image) that were obtained through crowd sourcing.
- The Flickr Audio Caption Corpus (Harwath and Glass, 2015), augments Flickr8K with spoken captions read aloud by crowd workers.
- F30kEnt-JP (Nakayama et al., 2020) provides Japanese translations of the captions (generated by humans). It covers the images and captions from Flickr30k (Young et al., 2014), a superset of Flickr8K, but only provides the translations of two captions per image.[1]

In all experiments, we use English as the source language for our models. While English is not a low-resource language, it is the only one for which we have spoken captions. The low-resource setting with translations is thus a simulated setting.

To summarize, we have 8,000 images with 40,000 captions (five per image), in both English written and spoken form (amounting to ∼34 hours of speech). In addition, we have Japanese translations for two captions per image.

Validation and test sets are composed of 1,000 images from the original set each (with corresponding captions), using the split introduced in Karpathy and Fei-Fei (2015). The training set is composed of the 6,000 remaining images.

We additionally introduce a smaller version of the dataset available for experiments with English transcriptions (later referred to as the *reduced* English dataset), matching in size the one used for experiments with Japanese translations (i.e. keeping only the sentences that have a translation, even though we use transcriptions).

[1]Items from F30kEnt-JP and Flickr8K were matched based on exact matches between the English written captions in both datasets. We also corrected for missing hyphens (e.g. "red haired" and "red-haired" are considered the same), leaving us with 15,498 captions with Japanese transcription.

3.4.2 Pre-processing

Image features are extracted from the pre-classification layer of a frozen ResNet-152 model (He et al., 2016) pretrained on ImageNet (Deng et al., 2009). We follow Merkx et al. (2019) and use features that are the result of taking the mean feature vector over ten crops of each image.

The acoustic feature vectors are composed of 12 MFCCs and log energy, with first and second derivatives, resulting in 39-dimensional vectors. They are computed over windows of 25 ms of speech, with 10 ms shift.

3.5 Repository

The code necessary to replicate our experiments is available under Apache License 2.0 at https://github.com/bhigy/textual-supervision.

4 Results

4.1 Impact of the architecture

We first look at the performance of the different models trained with the full Flickr8K training set and English transcriptions.

As expected, using directly text as input, instead of speech, makes the task much easier. This is exemplified by the difference between the *speech-image* and *text-image* models in Table 1.

Table 2 reports the performance of the four core models. We can notice that both pipeline and multitask architectures can use the textual supervision to improve results over the text-less baseline, though, pipeline approaches clearly have an advantage with a R@10 of 0.642. Unlike what one could expect, training the pipeline system in a sequential way does not bring any improvement over independent training of the modules (at least with this amount of data).

Comparing the two multitask approaches, we see that using ASR as a secondary task (*mtl-transcribe*) is much more effective than using text as another input modality (*mtl-match*).

Table 3 also reports performance of the best pipeline and the best multitask model on the test set. For completeness, Appendix A.2 reports the same results as Tables 1, 2 and 3 with different metrics (R@1, R@5 and Medr). Appendix A.3 reports the performance on the ASR task.

4.2 Using translations

Tables 1 and 2 report the performance of the same models when trained with Japanese transcriptions.

Model	English (full)	English (reduced)	Japanese
speech-image	0.416	0.280	0.285
text-image	0.702	0.653	0.626

Table 1: Validation set R@10 of our two reference models when trained either with English transcriptions, a reduced version of the English dataset, or Japanese translations.

Model	English (full)	English (reduced)	Japanese
pipe-ind	**0.642**	0.586	0.347
pipe-seq	**0.642**	**0.598**	0.345
mtl-transcribe/mtl-translate	0.569	0.478	**0.394**
mtl-match	0.451	0.356	0.337

Table 2: Validation set R@10 of our four core models, when trained either with English transcriptions, a reduced version of the English dataset, or Japanese translations.

Model	English	Japanese
pipe-seq	0.631	0.348
mtl-transcribe/mtl-translate	0.559	0.392

Table 3: Test set R@10 of the best pipeline (*pipe-seq*) and the best multitask (*mtl-transcribe/mtl-translate*) models, when trained with all English transcriptions or all Japanese translations.

The scores are overall lower than results with English transcriptions, which can be explained by two factors: (i) the size of the dataset which is only $\sim$2/5$^{\text{th}}$ of the original Flickr8K (as evidenced by the lower score of the text-less *speech-image* baseline) and (ii) the added difficulty introduced by the translation over transcriptions. Indeed, to translate speech, one first needs to recognize what is being said and then translate to the other language: thus translation involves many complex phenomena (e.g. reordering) which are missing from the transcription task.

While the four strategies presented in Table 2 improve over the *speech-image* baseline, their relative order differ from what is reported with English text. This time, the *mtl-translate* approach is the one giving the best score with a R@10 of 0.394, outperforming the pipeline systems (both performing similarly well in this context).

The difference in relative order of the models is likely the result of the degraded conditions (less data and harder task) impacting the translation task more severely than the speech-image retrieval task. The pipeline approaches, which rely directly on the output of the SLT component, are affected more strongly than the *mtl-translate* system where SLT is only a secondary target. This is in line with the results reported in the next section on downsampling the amount of textual data.

Table 3 reports performance of the best pipeline and the best multitask model on the test set. For completeness, Appendix A.2 reports the same results as Tables 1, 2 and 3 with different metrics (R@1, R@5 and Medr). Appendix A.3 reports the performance on the SLT task.

4.3 Disentangling dataset size and task factor

In an attempt to disentangle the effects of the smaller dataset and the harder task, we also report results on the reduced English dataset described in Section 3.4.1 (Table 1 and 2, 3$^{\text{rd}}$ column) . Looking first at Table 2, we can see that both factors do indeed play a role in the drop in performance, though not to the same extent. Taking *pipe-seq* model as example, reducing the size of the dataset results in a 7% drop in R@10, while switching to translations further reduces accuracy by 42%.

An unexpected result comes from the *text-image* system (Table 1). Even though the model works on ground-truth text (no translation involved), we still see a 4% drop in R@10 between the reduced English condition and Japanese. This suggests that the models trained with Japanese translations are not only penalized by the translation task being harder, but also that extracting meaning from Japanese text is more challenging than from English (possibly due to a more complicated writing system).

4.4 Downsampling experiments

We now report on experiments that downsample the amount of textual supervision available while keeping the amount of speech and images fixed.

We can see in Figure 3 (left) that, as the amount

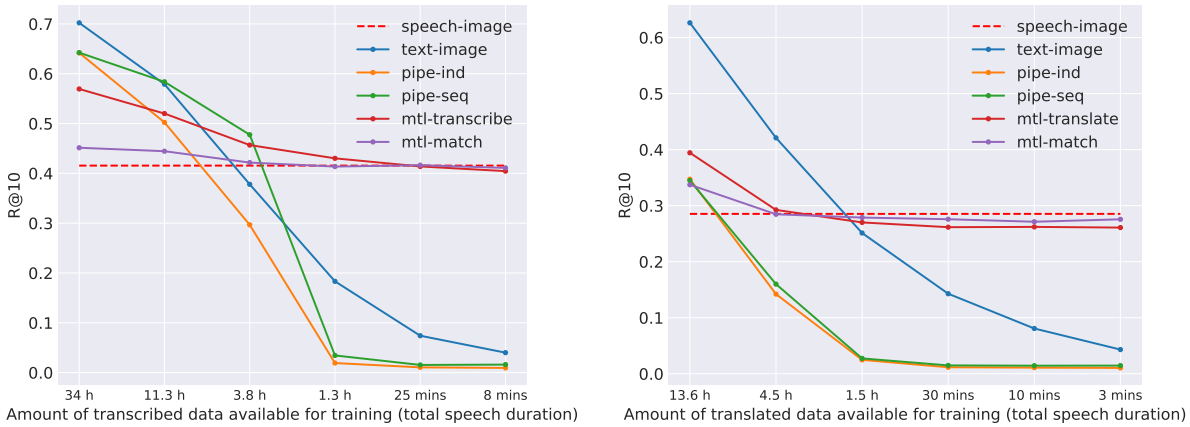


Figure 3: Results (R@10) of the different models on the validation set, when trained with decreasing amounts of English transcriptions (left) or Japanese translations (right). The total amount of speech available is kept identical, only the amount of translated data changes.

of transcribed data decreases, the score of the *text-image* and pipeline models progressively goes toward 0% of R@10. Between 11.3 and 3.8 hours of transcribed data, the *text-image* and *pipe-ind* models fall below the performance of the *speech-image* baseline. The *pipe-seq* is more robust and its performance stays higher (the best actually) until the amount of data goes below 3.8 hours of transcribed speech. After that the R@10 falls abruptly. This is likely the effect of the sequential training procedure allowing the *pipe-seq* to use all the speech available to train the text-image module (by transcribing it with the ASR module). Below 3.8 hours of speech, the quality of the transcriptions given by the ASR system deteriorates to the point that it is not usable anymore by the downstream component.

The two multitask approaches, on the other hand, progressively converge toward the speech-only baseline. The *mtl-transcribe* approach is overall giving better results than the *mtl-match* approach but fails to give a significant advantage over other sytems. It is only after the performance of the *pipe-seq* system abruptly decreases (from 1.3 hours of transcribed speech and below) that the *mtl-transcribe* system can surpass this one, at which point it is already performing very close to the *speech-image* baseline.

Figure 3 (right) reports on the same set of experiments with Japanese translations. In this case too, the *text-image*, *pipe-ind* and *pipe-seq* models go toward 0% of R@10 as the amount of translated data decreases, while the *mtl-translate* and *mtl-match* systems converge toward the *speech-image* baseline. It seems though that 4.5 hours of translated data is not enough to see an improvement over the *speech-image* baseline with any of the models.

In this case, the *pipe-seq* model does not have a significant advantage over the *pipe-ind* model, likely due to the difficulty of the translation task. The same reason probably explains why the *mtl-transcribe* strategy is performing the best on the full dataset (as reported in Section 4.2). However, while the pipeline architectures never surpass the *mtl-translate* model in the experiments reported, it may be the case with more data.

For completeness, Appendix A.3 reports the performance of on the ASR and SLT tasks themselves, for decreasing amounts of textual data.

5 Conclusion

In this paper, we investigated the use of textual supervision in visually-grounded models of spoken language understanding. We found that the improvements reported in Chrupała (2019) and Pasad et al. (2019) are a low estimate of what can be achieved when textual labels are available. Among the different approaches we explored, the more traditional pipeline approach, trained sequentially, is particularly effective and hard to beat with end-to-end systems. This indicates that text is a very powerful intermediate representation. End-to-end approaches tend to perform better only when the amount of textual data is limited.

We have also shown that written translations are a viable alternative to transcriptions (especially for unwritten languages), though more data might be useful to compensate for the harder task.

5.1 Limitations and future work

We ran our experiments on Flickr8K dataset, which is a read-speech dataset. We are thus likely underestimating the advantages of end-to-end approaches over pipeline approaches, in that they can use information present in speech (such as prosody) but not in text. Running experiments on a dataset with more natural and conversational speech could show end-to-end systems in a better light.

On the other end, we restricted ourselves to training the ASR and NLU components of the pipeline systems independently. Recent techniques such as Gumbel-Softmax (Jang et al., 2017) or the straight-through estimator (Bengio et al., 2013) could be applied to train/finetune this model in an end-to-end fashion while still enforcing a symbolic intermediate representation similar to text.

In the same vein, it would be interesting to explore more generally whether and how an inductive bias could be incorporated in the architecture to encourage the model to discover such kind of symbolic representation naturally.

Acknowledgements

Bertrand Higy was supported by a NWO/E-Science Center grant number 027.018.G03.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proc. of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA. ArXiv: 1409.0473.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. *arXiv:1308.3432 [cs]*. ArXiv: 1308.3432.

Alexandre Bérard, Olivier Pietquin, Laurent Besacier, and Christophe Servan. 2016. Listen and Translate: A Proof of Concept for End-to-End Speech-to-Text Translation.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75. Publisher: Springer.

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, Doha, Qatar. Association for Computational Linguistics.

Grzegorz Chrupała. 2019. Symbolic Inductive Bias for Visually Grounded Learning of Spoken Language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6452–6462, Florence, Italy. 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, Vancouver, Canada. Association for Computational Linguistics.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. ISSN: 1063-6919.

Spandana Gella, Rico Sennrich, Frank Keller, and Mirella Lapata. 2017. Image Pivoting for Learning Multilingual Multimodal Representations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2839–2845, Copenhagen, Denmark. Association for Computational Linguistics.

Parisa Haghani, Arun Narayanan, Michiel Bacchiani, Galen Chuang, Neeraj Gaur, Pedro Moreno, Rohit Prabhavalkar, Zhongdi Qu, and Austin Waters. 2018. From Audio to Semantics: Approaches to End-to-End Spoken Language Understanding. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 720–726. ISSN: null.

D. Harwath and J. 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, Galen Chuang, and James 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, Calgary, AB. IEEE.

David Harwath, Antonio Torralba, and James Glass. 2016. Unsupervised Learning of Spoken Language with Visual Context. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1858–1866. Curran Associates, Inc.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, Seattle, WA, USA. IEEE.

M. Hodosh, P. Young, and J. Hockenmaier. 2013. Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics. *Journal of Artificial Intelligence Research*, 47:853–899.

Wei-Ning Hsu, David Harwath, and James Glass. 2019. Transfer Learning from Audio-Visual Grounding to Speech Recognition. *Proc. Interspeech 2019*, pages 3242–3246.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparametrization with Gumble-Softmax. In *Proc. of the International Conference on Learning Representations (ICLR)*.

Ákos Kádár, Desmond Elliott, Marc-Alexandre Côté, Grzegorz Chrupa\la, and Afra Alishahi. 2018. Lessons Learned in Multilingual Grounded Language Learning. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 402–412.

Herman Kamper and Michael Roth. 2018. Visually Grounded Cross-Lingual Keyword Spotting in Speech. In *Proc. The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages*, pages 253–257.

Andrej Karpathy and Li Fei-Fei. 2015. Deep Visual-Semantic Alignments for Generating Image Descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proc. of the 3rd International Conference for Learning Representations*, San Diego, CA, USA. ArXiv: 1412.6980.

Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. 2019. Speech Model Pre-Training for End-to-End Spoken Language Understanding. In *Proc. Interspeech 2019*, pages 814–818.

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

Hideki Nakayama, Akihiro Tamura, and Takashi Ninomiya. 2020. A Visually-Grounded Parallel Corpus with Phrase-to-Region Linking. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4197–4203, Marseille, France. European Language Resources Association.

Ankita Pasad, Bowen Shi, Herman Kamper, and Karen Livescu. 2019. On the Contributions of Visual and Textual Supervision in Low-Resource Semantic Speech Retrieval. In *Proc. Interspeech 2019*, pages 4195–4199.

Leslie N. Smith. 2017. Cyclical Learning Rates for Training Neural Networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472, Santa Rosa, CA, USA. IEEE.

Gabriel Synnaeve, Maarten Versteegh, and Emmanuel Dupoux. 2014. Learning words from images and speech. In *In NIPS Workshop on Learning Semantics*. Citeseer.

Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-Sequence Models Can Directly Translate Foreign Speech. In *Interspeech 2017*, pages 2625–2629. ISCA.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Component	Number of layers
text decoder	{1, **2**, 3}
speech encoder of the ASR module	{3, 4, 5, 6}
speech encoder of the SLT module	{4, **5**, 6}
speech encoder of the *mtl-transcribe* model	{(2, 2, 2), (3, 1, 1), (3, 2, 2), (4, 0, 0), (4, 0, 1), (4, 1, 0), **(4, 1, 1)**, (4, 1, 2), (4, 2, 1), (4, 2, 2), (5, 0, 0), (5, 1, 1)}
speech encoder of the *mtl-translate* model	{(3, 1, 1), **(4, 1, 1)**, (4, 2, 2), (5, 1, 1), (5, 2, 2), (6, 1, 1)}

Table 4: List of values we experimented with for the number of GRU layers in the different components. The best configuration is indicated in bold face.

Component	Dimension of the hidden state
speech-image model	{256, 512, 768, **1024**}
ASR module	{512, **768**, 1024}
mtl-transcribe model	{512, **768**, 1024}

Table 5: List of values we experimented with for the number of GRU layers in the different components. The best configuration is indicated in bold face.

A Appendices

A.1 Choice of hyperparameters

The hyperparameters related to the optimization procedure and the architecture of the *speech-image* model where chosen based on Merkx et al. (2019). While the architecture of the other components is greatly inspired by this baseline, the number of GRU layers and the dimension of their hidden state were manually tuned to optimize accuracy. An exception to this is the *mtl-match* model for which the number of layer of the speech and text encoders is taken from Chrupała (2019). Optimization is done based on single runs.

A.1.1 Number of GRU layers

Table 4 reports the values we experimented with for the number of GRU layers in the text encoder, the speech encoder of the ASR module and the speech encoder of the SLT module. For the *mtl-transcribe* and *mtl-translate* systems, we report the triplet corresponding to the number of the layers in the shared encoder, the encoder dedicated to the speech-image task and the encoder dedicated to the transcription/translation task.

A.1.2 Dimension of the hidden state of the GRU layers

Table 5 reports the values we experimented with for the dimension of the hidden state of the GRU layers in the *speech-image* model, the ASR component and the *mtl-transcribe* model. The best value for the *speech-image* model was reused for the *text-image* and *mtl-match* models, as well as the text-image component of the pipeline models. The best value for the ASR module and the *mtl-transcribe* model was reused for the SLT module and the *mtl-translate* model.

A.2 Complete set of resutls

We report here on the performance of the models presented in section 4 (Tables 1, 2 and 3) with additional metrics, namely R@1, R@5 and Medr. Tables 6 and 7 report the performance on the validation set of the two reference and the four core models respectively. Table 8 reports the performance on the test set of the best pipeline and multitask models. Performance appears consistent accross metrics.

A.3 Performance of the ASR and SLT systems

Tables 9 and 10 respectively report the performance of the ASR and SLT modules on their own tasks, when trained on decreasing amount of textual data. Evaluation is performed on the validation set with a beam of width 10.

Model	English (full)			English (reduced)			Japanese		
	R@1	R@5	Medr	R@1	R@5	Medr	R@1	R@5	Medr
speech-image	0.105	0.299	16.2	0.059	0.188	38.3	0.059	0.192	36.0
text-image	0.258	0.566	4.0	0.228	0.508	5.0	0.209	0.492	6.0

Table 6: Validation set performance (R@1, R@5 and Medr) of our two reference models when trained either with English transcriptions, a reduced version of the English dataset, or Japanese translations.

Model	English (full)			English (reduced)			Japanese		
	R@1	R@5	Medr	R@1	R@5	Medr	R@1	R@5	Medr
pipe-ind	0.232	0.514	5.0	0.187	0.452	7.0	0.088	0.248	27.3
pipe-seq	0.224	0.509	5.2	0.190	0.459	7.0	0.082	0.242	27.0
mtl-transcribe/mtl-translate	0.177	0.431	7.8	0.133	0.352	12.0	0.091	0.285	19.3
mtl-match	0.115	0.321	13.0	0.079	0.244	24.3	0.071	0.232	26.3

Table 7: Validation set performance (R@1, R@5 and Medr) of our four core models, when trained either with English transcriptions, a reduced version of the English dataset, or Japanese translations.

Model	English			Japanese		
	R@1	R@5	Medr	R@1	R@5	Medr
pipe-seq	0.218	0.499	6.0	0.079	0.248	26.5
mtl-transcribe/mtl-translate	0.174	0.425	8.0	0.099	0.279	19.0

Table 8: Test set performance (R@1, R@5 and Medr) of the best pipeline (*pipe-seq*) and the best multitask (*mtl-transcribe/mtl-translate*) models, when trained with all English transcriptions or all Japanese translations.

Amount of transcribed data	34 h	11.3 h	3.8 h	1.3 h	25 mins	8 mins
Word error rate	0.154	0.238	0.397	0.801	1.034	0.977

Table 9: Performance (WER) of the ASR component on the validation set, when trained with decreasing amount of transcribed data.

Amount of translated data	13.6 h	4.5 h	1.5 h	30 mins	10 mins	3 mins
BLEU score	0.256	0.153	0.073	0.065	0.040	0.021

Table 10: Performance (BLEU score) of the SLT component on the validation set, when trained with decreasing amount of translated data.