Can We Use Small Models to Investigate Multimodal Fusion Methods?

Lovisa Hagström[1] **Tobias Norlund**[1,2] **Richard Johansson**[1,3]
[1]Chalmers University of Technology, [2]Recorded Future,
[3]University of Gothenburg
{lovhag, tobiasno, richajo}@chalmers.se

Abstract

Many successful methods for fusing language with information from the visual modality have recently been proposed and the topic of multimodal training is ever evolving. However, it is still largely not known what makes different vision-and-language models successful. Investigations into this are made difficult by the large sizes of the models used, requiring large training datasets and causing long train and compute times. Therefore, we propose the idea of studying multimodal fusion methods in a smaller setting with small models and datasets. In this setting, we can experiment with different approaches for fusing multimodal information with language in a controlled fashion, while allowing for fast experimentation. We illustrate this idea with the math arithmetics sandbox. This is a setting in which we fuse language with information from the math modality and strive to replicate some fusion methods from the vision-and-language domain. We find that some results for fusion methods from the larger domain translate to the math arithmetics sandbox, indicating a promising future avenue for multimodal model prototyping.

1 Introduction

Having models learn language from text alone has been criticised based on several aspects, from fundamental arguments about how language works (Bender and Koller, 2020; Bisk et al., 2020) to findings on certain information lacking in text (Gordon and Van Durme, 2013; Paik et al., 2021). Consequently, there is much interest in creating models that learn from more than text, i.e. "multimodal models". Many different multimodal models that fuse different types of information with text have been developed, ranging from vision-and-language models (VL models) (Zhang et al., 2020) to language models fused with knowledge graphs (Yu et al., 2022). In this work, we mainly focus on the vision-and-language domain, while our results may generalize to other multimodal domains as well.

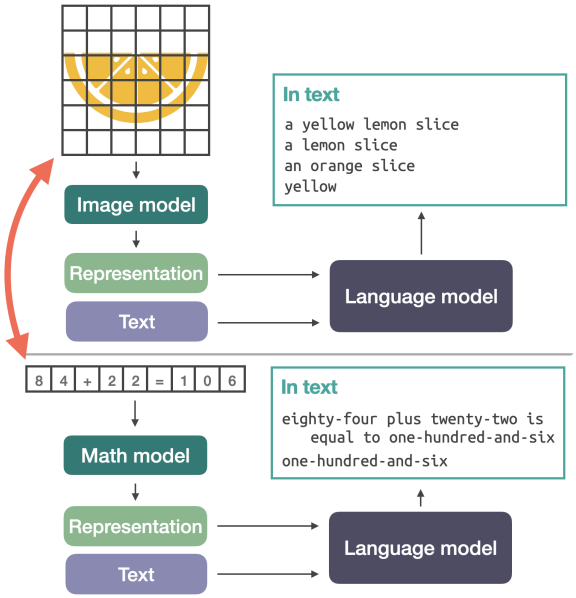


Figure 1: An overview of the vision-and-language fusion method and our proposed math-language fusion method. The image model and the math model are backbones.

A standard approach for fusing information from images with language is firstly to use a *backbone*, a large neural network that has been trained to create good representations of images. One such model is ResNet trained on Visual Genome (Anderson et al., 2018; Krishna et al., 2016). The training data in this standard approach typically consists of image samples paired with linguistic information, such as MS COCO (Lin et al., 2014). Multimodal fusion is then achieved by generating representations of image samples using the backbone and feeding those together with paired linguistic samples to a large language model pre-trained on text. The language model is then expected to learn to leverage the visual information for its linguistic usage if trained on a sufficient amount of image-text pairs. We illustrate this approach in Figure 1.

Significant success on several multimodal vision-and-language tasks has been achieved with this ap-

Proceedings of the 2022 CLASP Conference on (Dis)embodiment, pages 45–50
September 15–16, 2022.

proach (Chen et al., 2020; Li et al., 2019, 2020; Tan and Bansal, 2019). However, it has seldom been investigated from a methodological perspective, and existing methodological investigations indicate that performance on different vision-and-language tasks may not originate from what we would expect. For example, differences in performance for different VL models have previously been ascribed to different model architectures, while Bugliarello et al. (2021) recently showed that training data and embedding layers matter more than e.g. if the model is dual- or single-stream. Hessel and Lee (2020) also showed that performance improvements for some VL models on image-text classification tasks mainly originate from unimodal signals, and not cross-modal interactions. Additionally, Frank et al. (2021) found that VL models are not necessarily symmetrical in their cross-modal interactions.

Additionally, the standard approach for fusing information from images with language does not necessarily encourage experiments with different methods for multimodal fusion, mainly since the associated compute power is substantial, as a consequence of large models and data sizes. For example, the size of the Faster R-CNN visual features used by Bugliarello et al. (2021) is approximately 1.6 TB, significantly larger than e.g. the 20.3 GB for English Wikipedia[1].

In this work, we hypothesize that methods for multimodal fusion can be developed in a smaller domain for more efficient investigations. We experiment with a *sandboxed* experimental setting for investigating different multimodal fusion methods. It is based on synthetic multimodal data, in a very constrained math-and-language domain consisting of simple arithmetic math statements, as illustrated in Figure 1. Using this setup, we are free to investigate different multimodal fusion methods like the one seen in Figure 1 *in silico*, using models that are easier to work with, with faster training times and full control of the data since we can synthetically generate it. The sandbox is described in Section 2 and we release the corresponding code to enable other researchers to build on it.[2]

We reason that results in the math-and-language domain could generalize to more complex domains, such as the vision-and-language domain (Section 2) and provide empirical support for this with a few experiments (Section 3).

To summarize, our contributions are two, 1) proposal of idea in using smaller domains for easier investigations of vision-and-language fusion methods, and 2) demonstration of idea in the math arithmetics sandbox set in the math-and-language domain. We couple this demonstration with validating experiments that compare against recent results in the vision-and-language domain.

2 The math arithmetics sandbox

Similarly to how images are described with pixel values over some grid, equations can be described with numerical values over potential positions in an equation. Also similarly to how we use image specific models to generate representations of images for language fusion, we can use math specific models to generate representations of equations for language fusion, illustrated in Figure 1.

In the math arithmetics sandbox we limit ourselves to two-digit numbers and equations describing sums of these numbers. An example of a data sample in the math modality of the sandbox is then $54 + 21 = 75$, with the corresponding string "fifty-four plus twenty-one is equal to seventy-five". We can work with a total of $10^4 = 10,000$ different such math-language pairs in our sandbox. This number of possible samples is much smaller than e.g. the corresponding number for image data, for which the size of the support is $(V^3)^{32\times32}$ for 32×32-pixel images with V possible values for each RGB channel, with underlying probability distributions of the pixel values.

An example of a potential task in the math arithmetics sandbox is to predict the continuation of the string "fifty-four plus twenty-one is equal to" given the math information $54 + 21 = 75$. This task can be compared to the task of visual question answering in the vision-and-language domain, for which a question could be "What is the color of the fruit?" provided with an image of a yellow lemon. The underlying format of the two tasks is essentially the same, a text prompt is provided together with information from another modality that encodes the information necessary to answer the prompt.

We hypothesize that results from experiments in the math arithmetics sandbox can give intuitions about the vision-and-language domain, since the difference between the math arithmetics sandbox and more complex multimodal domains mainly lies in the size of the support, while the underlying for-

[1]Provided by Huggingface via https://huggingface.co/datasets/wikipedia

[2]Available at https://github.com/lovhag/small-math-language-multimodal-fusion

mat for the multimodal tasks are similar. Thus, it is interesting to investigate whether we can acquire valuable knowledge in the math arithmetics sandbox and exploit it in more complex multimodal domains, such as the vision-and-language domain.

3 Experiments

To investigate whether insights gained in the math arithmetics sandbox are comparable to other multimodal domains, we perform a set of experiments.

3.1 Setup

We validate our math arithmetics sandbox by comparing findings from Huang et al. (2021) and Bugliarello et al. (2021) with findings we obtain on the same, but sandboxed, cases. The findings we investigate and aim to reproduce are:

1. Adding information to a model from a modality with a sufficient amount of train samples improves on the performance of a model (Huang et al., 2021).
2. Training with an additional modality with too few training samples weakens the performance of a model compared to not adding the extra modality (Huang et al., 2021).
3. The design of the embedding layers matters for VL models (Bugliarello et al., 2021).
4. Dual- and single-stream VL model architectures perform on par (Bugliarello et al., 2021).

Task The task of the validating experiments is to accurately predict the answer to a text version of a sum equation x_T, given the corresponding complete math equation x_M. An input example for this task is $x_T =$ "one plus two is equal to", $x_M =$ `1+2=3` and with the correct answer $y =$ "three".

During validation, the multimodal model is to generate the continuation of an incomplete string equation given the corresponding complete math equation. For simplicity, we use a greedy decoding scheme and measure the $k = 1$ accuracy.

Models As illustrated in Figure 1 our problem setup firstly consists of a math model M_M that generates a representation $M_M(x_M)$ of a math input x_M. This representation is then given to a language model M_L together with the incomplete text input x_T to get a prediction $y' = M_L(x_T, M_M(x_M))$.

We model M_M with a small version of GPT2[3] (Radford et al., 2019) with a vocabulary restricted to only include the tokens in the math equations. The output of $M_M(x_M)$ is then a set of vector representations, one for each token in x_M.

M_L is also modelled with a model similar to GPT2. The difference here is that we need to adapt the model to accept a multimodal input (x_T and $M_M(x_M)$). For this, we design a special embedding layer and choose between a single- or dual-stream architecture for the model.

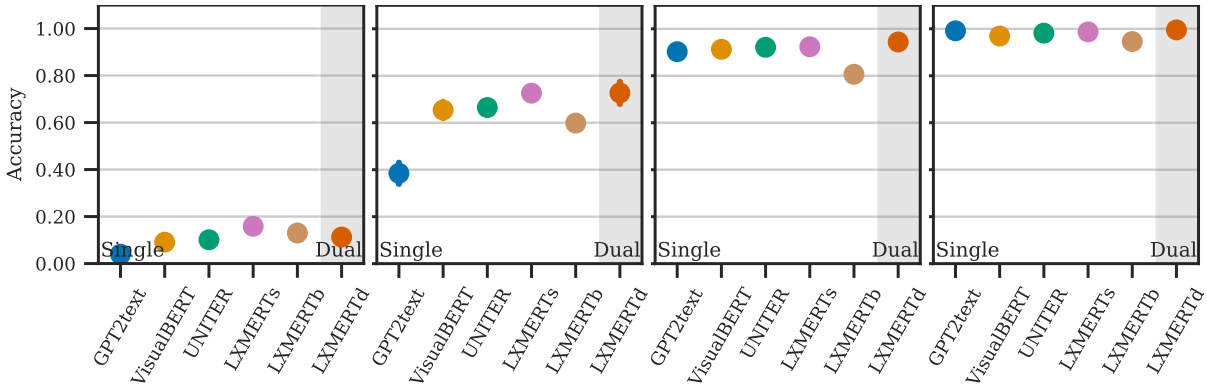
Since one goal of this article is to investigate the effect of different embedding layer designs and dual- versus single-stream model architectures, we create and evaluate four different model variations, seen in Table 1. The embedding designs essentially determine how the multimodal model input x_T and $M_M(x_M)$ is processed before being passed to the encoder of M_L. The dual- or single-stream architectures essentially determine how early linguistic information and visual information is fused while being processed in M_L. We name the model variations after their embedding design, stream type and amount of training data for the math model (backbone). The embedding designs are named after the VL model that incorporates the same design. Detailed descriptions of how we create the model variations can be found in Appendix A.

Name	Embedding	Stream	Backb
GPT2text			
VisualBERT	VisualBERT	Single	99%
UNITER	UNITER	Single	99%
LXMERTs	LXMERT	Single	99%
LXMERTd	LXMERT	Dual	99%
LXMERTb	LXMERT	Single	5%

Table 1: Backb denotes the backbone, which was trained on either 99% or 5% of the math data. GPT2text is a standard unimodal GPT2 model only taking text input, serving as a unimodal baseline.

Data The data we have available are the 10,000 math-language pairs. We train the math model M_M on 9,900 (99%) of the pure math equations samples such that it attains a validation accuracy of 0.99, to ensure that the model is able to generate information-rich math features. We then train the language model M_L on math-text pairs. We experiment with different sizes of training data and evaluate on the remaining data samples, to investigate the effect of having access to many or few text samples. We train the models until they have

[3]Embedding dimension of 64, 4 Transformer layers, inner dimension of 256 and 8 attention heads.



(a) 100 samples (b) 500 samples (c) 1000 samples (d) 2000 samples

Figure 2: The validation accuracies for the different sandboxed models trained on 100, 500, 1000 and 2000 math-text samples respectively. The validation task was to predict the answer to a string version of a sum equation. Each configuration was trained 5 times.

converged in performance on the validation set and report their accuracy score on the same set.

We also train a math model on only 500 samples (5%) of the pure math equations to a validation accuracy of 0.67. We use this backbone to investigate 2) the effect of training with too few samples for the additional modality and train a single-stream model with LXMERT embeddings with this backbone, denoted 'LXMERTb'.

3.2 Results

The results from the validating experiments are shown in Figure 2.

Does adding multimodal information improve model representations? Yes, for all cases in which the language data is more scarce (<2000 samples), adding multimodal information leads to a better model performance than only using text.

Does training with insufficient data for the additional modality weaken model performance? Predominantly yes, when there is more text data (500< samples) adding information from the "bad" math backbone has a deteriorating impact on the model performance and the multimodal model even performs worse than the unimodal version. When there is less paired math-text data, additional information from the backbone trained on little data leads to a better performance than in the pure-text case. Potentially, this is due to the pure text case also having too little data, such that additional information, despite bad quality, still is helpful.

Do embeddings matter for our models? No, we do not observe significantly different performances for the models with different embedding layers. Potentially, this is due to the features from the math modality being so simple that the embedding does not really matter, compared to the case of e.g. features from an object detector that also encode locations of bounding boxes.

Do dual- and single-stream architectures perform on par? Yes, despite the fact that the sandboxed dual-stream model is larger than the single-stream models, the model performs on par with the single-stream models. This is in agreement with results found on the vision-and-language domain.

4 Conclusions

We propose the idea of studying vision-and-language fusion methods from a smaller domain. We reason that it would be easier and less resource demanding to develop performant multimodal fusion methods if we could investigate them using small models and domains. We exemplify this with the math arithmetics sandbox and get promising results. However, more experiments on other small domains are necessary to evaluate the potential of our proposed idea. It would also be beneficial to investigate small domains that are more similar to e.g. the vision-and-language domain.

Acknowledgements

We would like to thank the anonymous reviewers of this paper for their valuable feedback. Additionally, this work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. 2020. Experience grounds language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8718–8735, Online. Association for Computational Linguistics.

Emanuele Bugliarello, Ryan Cotterell, Naoaki Okazaki, and Desmond Elliott. 2021. Multimodal pretraining unmasked: A meta-analysis and a unified framework of vision-and-language BERTs. *Transactions of the Association for Computational Linguistics*, 9:978–994.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. UNITER: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer.

Stella Frank, Emanuele Bugliarello, and Desmond Elliott. 2021. Vision-and-language or vision-for-language? on cross-modal influence in multimodal transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9847–9857, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 25–30.

Jack Hessel and Lillian Lee. 2020. Does my multimodal model learn cross-modal interactions? it's harder to tell than you might think! In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 861–877, Online. Association for Computational Linguistics.

Yu Huang, Chenzhuang Du, Zihui Xue, Xuanyao Chen, Hang Zhao, and Longbo Huang. 2021. What makes multi-modal learning better than single (provably). *Advances in Neural Information Processing Systems*, 34.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. VisualBERT: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXX*, volume 12375 of *Lecture Notes in Computer Science*, pages 121–137. Springer.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Cory Paik, Stéphane Aroca-Ouellette, Alessandro Roncone, and Katharina Kann. 2021. The World of an Octopus: How Reporting Bias Influences a Language Model's Perception of Color. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 823–835, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.

Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. 2022. A survey of knowledge-enhanced text generation. *ACM Computing Survey (CSUR)*.

Chao Zhang, Zichao Yang, Xiaodong He, and Li Deng. 2020. Multimodal intelligence: Representation learning, information fusion, and applications. *IEEE Journal of Selected Topics in Signal Processing*, 14(3):478–493.

A Language Model variations

To investigate the effect of different vision-and-language fusion methods, we implement versions

of these in the model M_L in the math-and-language domain.

A.1 Design of embedding layers

We wish to investigate the effect of different embedding layer designs for text x_T and math features $M_M(x_M)$. The embedding designs essentially determine how the multimodal model input x_T and $M_M(x_M)$ is processed before being passed to the encoder of M_L. Similarly to the work by Bugliarello et al. (2021), we switch between different embedding designs, for which we test designs similar to those used in VisualBERT, UNITER and LXMERT (Li et al., 2019; Chen et al., 2020; Tan and Bansal, 2019). The embedding designs we use mainly differ in how layer normalization and dropout are applied to the x_T and $M_M(x_M)$ inputs.

In the vision-and-language domain the embedding designs are more differentiated since they process positional information differently. As a consequence of not having any extra positional information in the math-and-language domain we cannot imitate this positional embedding design difference in the math arithmetics sandbox.

A.2 Dual- and single-stream models

We also wish to investigate dual- and single-stream multimodal models. The single-stream model is already provided by the standard GPT2 architecture in the sense that we give it the concatenation of the math and linguistic features of an math-text pair as input, to allow for information fusion from the start. This design is similar to that of the single-stream VisualBERT model.

The dual-stream architecture we use is based on the dual-stream LXMERT architecture, in which we switch BERT specific layers for the corresponding GPT2 layers, such that the math and linguistic features are first processed by two independent stacks of Transformer layers before they are fed into cross-modal Transformer layers. Dual-stream models are typically larger, where e.g. LXMERT with 228M trainable parameters is two times larger than the 110M parameters of single-stream VisualBERT. For our sandboxed dual-stream model, we go by the same principle and set the number of linguistic Transformer layers to three, the number of relational layers to two and the number of cross-attention layers to two, such that we get a sandboxed dual-stream model that is approximately two times larger than the sandboxed single-stream models, corresponding to the size difference between VisualBERT and LXMERT.

The number of trainable parameters for the sandboxed single-stream models is approximately 211K. For the sandboxed dual-stream model the number of trainable parameters is 495K.