Are we asking the right questions in MovieQA?

Bhavan Jasani[1] Rohit Girdhar[1] Deva Ramanan[1,2]
[1]The Robotics Institute, Carnegie Mellon University [2]Argo AI
http://BhavanJ.github.io/MovieQAWithoutMovies

Abstract

Joint vision and language tasks like visual question answering are fascinating because they explore high-level understanding, but at the same time, can be more prone to language biases. In this paper, we explore the biases in the MovieQA dataset and propose a strikingly simple model which can exploit them. We find that using the right word embedding is of utmost importance. By using an appropriately-trained word embedding, about half the Question-Answers (QAs) can be answered by looking at the questions and answers alone, completely ignoring narrative context from video clips, subtitles, and movie scripts. Compared to the best published papers on the leaderboard, our simple question+answer only model improves accuracy by 5% for video + subtitle category, 5% for subtitle, 15% for DVS and 6% higher for scripts.

1. Introduction

Language has long been an integral part of visual understanding. From objects [4, 11] to human actions [8], categorization of visual data has lead to rapid developments in computer vision. However, language is particularly transformative because it can be applied to domains beyond simple classification, such as image captioning [23] and Visual Question-answering (VQA) [1]. Indeed, VQA has arguably emerged as a now-standard vision task, primarily due to its flexibility and standardized evaluation.

MovieQA: QA tasks are particularly intriguing for videos, where they can explore cognitive storytelling concepts (such as intentions and goals) difficult to extract from static images. Unsurprisingly, there have been considerable efforts in bridging the gap between language and spatio-temporal understanding of videos [10, 19]. To that end, a recently released dataset, MovieQA [19], extends the VQA philosophy to videos, by collecting short real-world movie clips, along with subtitles and wiki-plots, and defining multiple choice questions on them. It has 5 categories for the QA task based on the information used: 1) movie clips + subtitles 2) movie subtitles 3) movie scripts 4) DVS (descriptive video services) 5) Wikipedia movie plots (wiki-plots). The first category is based on the combination of visual and text data, whereas the remaining 4 are purely text-based tasks. While there has been a significant amount of work in this direction, most methods [9, 12, 15, 22] do not make strong use of visual features and instead rely heavily on language-based cues such as subtitles or wiki-plots. This raises the question: are our video models unequipped to truly understand videos, or is the MovieQA task unfairly biased against actually needing visual information?

WikiWord embeddings: In this work, we explore this question in detail. We propose a strikingly simple approach that extracts average-pooled word embeddings of the question and each answer and reports the answer with the best correlation. We train our word embedding model – named *WikiWord* embeddings – on unsupervised Wikipedia plots, to capture the narrative structure of movie plots. We find that this simple model outperforms *all* reported methods on MovieQA [19] test set. This includes models that use subtitles, scripts, and videos, while our naive model uses *only* the question and answer. We have submitted our results to the test evaluation server, and are ranked first in four out of five categories at the time of submission of this paper.

The role of plots: It is worth noting the one category that we do *not* win is plot-synopsis (wiki-plots), where the current state-of-the-art is quite high (85%). This is explained by the fact that the question and answers were *constructed* by inspection of movie plots from Wikipedia. This category provides aligned training examples of $\{(\text{question,answer,plot})_i\}$ tuples for supervised learning, which can be exploited by powerful language models that are trained on such aligned data [3]. In contrast, we learn embeddings in an unsupervised fashion from *un*aligned movie plots $\{\text{plot}_i\}$. This information is freely available in all the 5 benchmark category protocols. Our results demonstrate that *unsupervised* learning of word-embeddings from *un*aligned movie plots still captures a rich amount of narrative structure about the movies of interest.

Source of bias: The source of language bias might be explained by the procedure used to generate the benchmark QAs: Amazon Turkers generate candidate QAs by reading

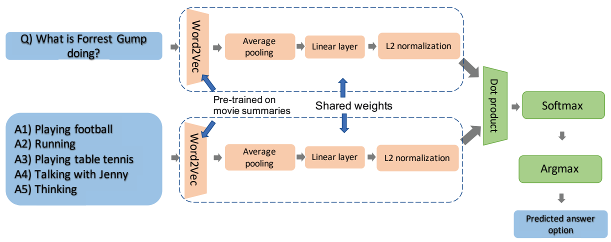

Figure 1: **WikiWord Embedding model.** It takes as input the question and 5 answer choices. For every word in the sentence (of question and answer choices), a 300D word embedding is computed using word2vec. This word2vec is pretrained on movie plots and its weights are kept fixed. The word embedding is average pooled to get a sentence level vector and then passed through a linear layer (initialized as an identity matrix) to get another 300D vector which is then L2 normalized. Dot product similarity is computed for the 300D representation of question and the 5 answer choices, and the one with the highest value is picked as the model's predicted answer option.

the movie plots *without* watching the movies. Movie clips are later programmatically aligned to movie plot lines and the questions. Moreover, we find that for many QAs, words and characters from the relevant movie plots are included in the correct answer, but not included in the incorrect answers. This may make it easier to pick out the correct answer by simply looking at the question and answers.

Why is this relevant for vision? Because our central technical contribution is a novel language baseline model, one might argue that it is not relevant for a vision audience. We believe it is crucial to ensure that strong baselines are introduced for the tasks at hand, to ensure meaningful progress is made. Hence, we feel that our results are very relevant for the MovieQA community and future joint language-vision datasets. Additionally, our method naturally generates a partition of the data that is free of such trivial biases and can potentially be used for further progress in video-language modeling.

2. Related Work

Video and language: Joint learning of language and vision has been explored in various ways. This includes movie descriptions [18], video understanding through fill in the blank [13], video retrieval [20], character co-referencing [17] and image captioning [23]. Most previous works have focused on using movies [6, 17, 20], because they provide time synchronized audio, subtitles and videos.

Visual QA task: Question answering provides an easy and unambiguous evaluation metric for joint language and vision tasks. The task is to predict the correct answer from a list of options for a given question based on a story, which provides the context. Many visual question answering datasets have been recently released, including image-based question answering datasets like VQA [1], and more recently, video-based QA. This includes datasets like MovieQA [19], constructed from movies, TVQA [10], constructed from TV series and TGIF QA [7], constructed from GIFs. Additionally, there has been work on reading comprehension [5], which are the purely language-based QA datasets.

3. Our Approach: WikiWord Embeddings

Classic formulations: Typical QA task can be formalized as triplets consisting of the reference passage (to be comprehended), a question, and the possible answers (5 choices in case of MovieQA). Contemporary QA systems create a scoring function that iterates over all putative answers, conditioned on the question and reference passage, returning the highest-scoring answer.

Default word2vec: Let us first review the basic Visual QA framework provided in the MovieQA benchmark [19], which forms the basis for our proposed solution. Of particular relevance is the default word2vec, which is trained on 1400 Wikipedia movie plots, including movies in the train split, test split, and movies outside MovieQA. It is important to note that the word embeddings are learnt from movie plots in an unsupervised way, without looking at the corresponding questions and ground-truth answers.

WikiWord embedding model: Our crucial modification trains a word2vec embedding *only* on movies present in MovieQA (train and test splits), a strict subset of the data used to train the default word2vec embedding. We call our embedding WikiWords. We use it in a simple pipeline (Fig. 1) that makes use of *only* questions and answers, ignoring any reference passage, subtitles, or videos. Specifically, we compute a sentence-level embedding for each question and answer by average pooling WikiWord embeddings. We then select the answer with the highest (weighted) similarity to the question. Note that the linear reweighting is the only component of our model that is trained on question-answer pairs. We also provide experimental results for a variant of our model without any linear tuning, which is trained without *any* question-answer supervision.

4. Experiments

Leaderboard results: The dataset is divided into train, validation (val) and test splits. We report ablation experiments on the val set. The test results are obtained from the official server[1], and are reported in Tables 1 and 2 for the various categories. Table 3 shows the performance of different input modalities (QA only, subtitles, videos and videos+subtitles) for the top model on the leaderboard with publicly released code [22]. Our results dominate past work by a significant

[1] http://movieqa.cs.toronto.edu/leaderboard/

Leader board submission	Subtitles
Our QA-only model	**44.01**
Speaker Naming in Movies [2]	39.36

Leader board submission	DVS
Our QA-only model	**49.65**
MovieQA benchmark [19]	35.09

Leader board submission	Scripts
Our QA-only model	**45.49**
Read Write Mem. Net. [15]	39.36

Table 1: MovieQA leaderboard for Subtitles, DVS, and Scripts categories at the time of submission along with the second best submissions.

Leader board submission	Movie: Video+Subtitles
Our QA only model	**46.98**
New method to optimize all MEM network (anonymous)	45.31
Multimodal dual attention memory [9]	41.41

Table 2: MovieQA leaderboard for Video+Subtitles category at the time of submission along with previous best anonymous and published results.

Modality	Google [14]	MovieQA [19]	Our best w2v
QA only	24.71	38.70	**50.00**
Subtitle	25.16	36.45	**47.62**
Video	27.87	36.45	**50.67**
Videos + subtitle	25.39	40.06	**48.87**

Table 3: Validation experiments with different input modalities and for different word embeddings on best model on MovieQA leaderboard with publicly released code, Layered Memory Network [22]. Using subtitles or videos does not improve accuracy. In general, performance differences due to input modalities are dwarfed by the benefits of a better word embedding.

margin (5 percent), while using strictly less information for learning word embeddings and ignoring reference material such as subtitles, DVS, or scripts.

4.1. Ablating the word embeddings

Movie specific words: We experimented with word2vec (w2v) trained on different data - 1) Google w2v (trained on 100 billion words from Google News dataset, has a vocabulary of 3 million words) 2) MovieQA w2v (provided by the authors, which is trained on about 1400 movie plot synopses including all 408 movies in the MovieQA dataset) 3) Our WikiWords, which is trained on train+test MovieQA plots. Figure 2 visualizes Google w2v and WikiWords. Google w2v is generic and may not contain the names of characters and entities in specific movies. On the other hand, WikiWords tends to embeds words from the same movie together - e.g. 'Quidditch' and 'Harry' refer to the movie Harry Potter. Hence WikiWords captures movie-specific semantics, which is very helpful in answering questions.

Google and MovieQA word2vec: Table 4 shows the performance of different w2v's with our QA-only model, evaluated on the train and val set (since submission to the online test server are limited). We plot performance both with and without fine-tuning our linear weighting layer. Google w2v performs poorly and close to chance (20%, second row), even after fine-tuning. This is likely because movie-specific words are missing in its vocabulary. MovieQA w2v itself gives about 38.71% accuracy (first row) after fine-tuning.

#	W2V	Movie plots for training w2v	Train accuracy (w/o fine-tune)	Train accuracy	Val accuracy (w/o fine-tune)	Val accuracy
1	[19]	Gen + train + val	27.70	41.67	26.74	38.71
2	[14]	Google News	17.84	30.40	14.56	20.31
3	Ours	Val	20.30	24.43	40.51	41.98
4	Ours	Train	40.19	57.46	18.39	19.30
5	Ours	Train+val	39.90	51.64	38.48	**49.88**
6	Ours	Gen	21.34	21.44	17.17	18.17
7	Ours	Gen+val	21.31	27.26	34.76	36.11
8	Ours	Gen+train	36.77	55.33	16.59	19.63
9	Ours	Gen+train+val	36.01	54.40	32.73	41.53

Table 4: Experiments with our QA only model (for movies+subtitle task) with different amount of movie plots used for training Word2Vec (W2V). This table shows the importance of different word embeddings. Generic word embeddings, like Google's (row 2) gives really poor accuracy. And using a better word embedding (row 5) can give really high accuracy, even without training the QA only model. When we use only val movie plots (row 3) we get good val accuracy but bad train accuracy and vice-versa. Highest accuracy is achieved when we use plots from train+val movies (row 5). Adding movies not in the dataset (row 9), results in degradation of accuracy. Even though same data are used for first and last row, the results differ because of slightly different hyper-parameters.

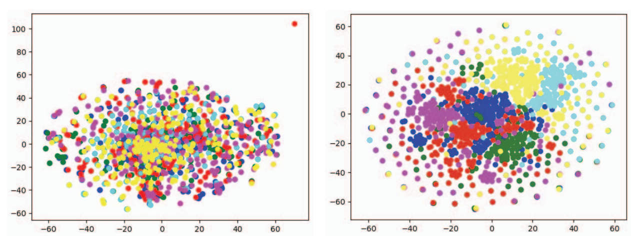

Figure 2: Left: t-SNE visualization of word embeddings based on google w2v; right: for our WikiWord Embedding w2v. We show them for words taken from 6 different movies, words from the same movies have the same color. For the generic word embedding like google w2v, words from different movies are all jumbled up together and hence they lose the movie-semantics important for this task. In WikiWords, the words from the same movie are clustered together and away from those from other movies.

Our word2vec: We now explore the effect of using subsets of movie plots to train w2v: 'train', 'val', and 'gen' refer to plots from the train, val, and 1400-(train+val) movies respectively. Table 4 shows that when including 'val', our QA-only model is able to get high accuracy (40.51%) even without fine-tuning. This is notable because this system is not trained on *any* question-answer pairs. Finally, Wikiwords (training w2v on 'train+val') leads to the best performance (49.88%). Hence, just using plots which are part of the dataset leads to the best accuracy and adding additional movie plots from the general population degrades performance.

Subtitle based word2vec: As another baseline, we train a w2v with subtitles instead of movie plots from 'train+val' movies and use it in our QA model. This leads to a low accuracy of 26.41% indicating that w2v trained on subtitles is not able to capture the semantics to exploit the language bias.

4.2. TVQA dataset experiments

It is worth exploring the performance of our WikiWord embedding more generally on other datasets. TVQA [10] is a recent video QA dataset collected from 6 TV series. In contrast to MovieQA, the Mechanical Turkers actually watched the videos (and also read the aligned subtitles) while generating the QAs for TVQA. Since there are no equivalent to movie plots for the TV series, we perform experiments by training word embedding with the subtitles for TVQA dataset. Results in Table 5 indicate that although about 40% of the QA's can be answered without using any context (this result is also mentioned by the TVQA authors), the nature of data used for training word embedding doesn't seem to affect the performance of the QA only models. This shows that TVQA dataset better controls for the biases that MovieQA has.

Model	Word embedding	Val accuracy
WikiWord embedding	Google News [14]	32.76
	TVQA subtitles	32.66
TVQA baseline [10]	Random weights	39.61
	Wikipedia GLOVE [16]	40.18
	TVQA subtitles	39.65

Table 5: Performance of two QA only models on TVQA dataset - 1) WikiWord embedding model 2) TVQA baseline model [10] proposed in the paper. For both the models we experiment with word embeddings trained from different data and observe that the performance doesn't change.

5. Conclusion

We show that the MovieQA dataset has language bias and present a simple QA only model that exploits it. Our key idea is to train the word2vec model on a *subset* of the data used by state of the art methods, by focusing only on the train and test movie plots. This model achieves state of the art performance on four of the five categories on the leaderboard at the time of submission.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *CVPR*, 2015. 1, 2

[2] M. Azab, M. Wang, M. Smith, N. Kojima, J. Deng, and R. Mihalcea. Speaker naming in movies. *arXiv preprint arXiv:1809.08761*, 2018. 3

[3] M. Blohm, G. Jagfeld, E. Sood, X. Yu, and N. T. Vu. Comparing attention-based convolutional and recurrent neural networks: Success and limitations in machine reading comprehension. *arXiv preprint arXiv:1808.08744*, 2018. 1

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*. Ieee, 2009. 1

[5] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. In *NeurIPS*, 2015. 2

[6] Q. Huang, Y. Xiong, Y. Xiong, Y. Zhang, and D. Lin. From trailers to storylines: An efficient way to learn from movies. *arXiv preprint arXiv:1806.05341*, 2018. 2

[7] Y. Jang, Y. Song, Y. Yu, Y. Kim, and G. Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, 2017. 2

[8] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1

[9] K.-M. Kim, S.-H. Choi, J.-H. Kim, and B.-T. Zhang. Multimodal dual attention memory for video story question answering. In *ECCV*, 2018. 1, 3

[10] J. Lei, L. Yu, M. Bansal, and T. Berg. Tvqa: Localized, compositional video question answering. In *EMNLP*, 2018. 1, 2, 4

[11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*. Springer, 2014. 1

[12] C.-N. Liu, D.-J. Chen, H.-T. Chen, and T.-L. Liu. A2a: Attention to attention reasoning for movie question answering. In *ACCV*. Springer, 2018. 1

[13] T. Maharaj, N. Ballas, A. Rohrbach, A. Courville, and C. Pal. A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering. In *CVPR*, 2017. 2

[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, 2013. 3, 4

[15] S. Na, S. Lee, J. Kim, and G. Kim. A read-write memory network for movie story understanding. In *CVPR*, 2017. 1, 3

[16] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 4

[17] A. Rohrbach, M. Rohrbach, S. Tang, S. Joon Oh, and B. Schiele. Generating descriptions with grounded and co-referenced people. In *CVPR*, 2017. 2

[18] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, and B. Schiele. Movie description. *IJCV*, 2017. 2

[19] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. Movieqa: Understanding stories in movies through question-answering. In *CVPR*, 2016. 1, 2, 3

[20] A. Torabi, N. Tandon, and L. Sigal. Learning language-visual embedding for movie understanding with natural-language. *arXiv preprint arXiv:1609.08124*, 2016. 2

[21] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

[22] B. Wang, Y. Xu, Y. Han, and R. Hong. Movie question answering: remembering the textual cues for layered visual contents. In *AAAI*, 2018. 1, 2, 3

[23] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 1, 2