Are we asking the right questions in MovieQA?

Bhavan Jasani
Carnegie Mellon University
bjasani@cs.cmu.edu

Rohit Girdhar
Carnegie Mellon University
rgirdhar@cs.cmu.edu

Deva Ramanan
Carnegie Mellon University
deva@cs.cmu.edu

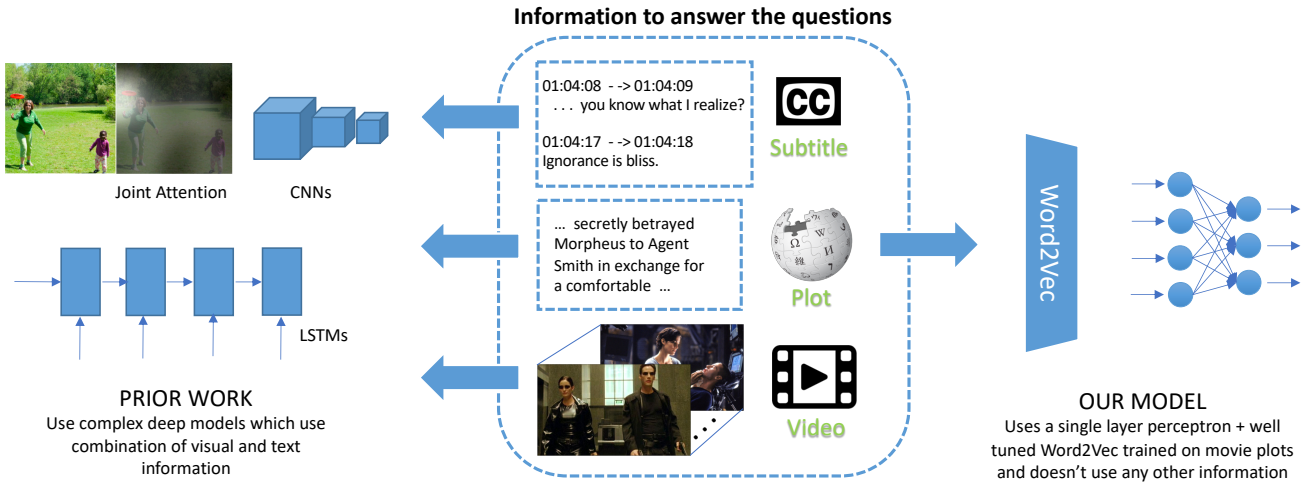

Figure 1: **Why watch a full movie when I can learn from Wikipedia movie plots?** The MovieQA task is: Given a question and multiple answer choices, find the correct answer by using the context provided in the corresponding videos and subtitles. Prior works use deep networks and claim to use information from videos and subtitles to do this question-answering task. We show a significantly simpler model that achieves state of the art accuracy. Our model uses a well tuned word embedding trained unsupervised on Wikipedia movie plots (which are movie summaries) and, does the question-answering task without using the information from videos or subtitles. Due to the language biases in question-answers, we found using a good word embedding provides enough information to answer about the half the questions in MovieQA dataset by just looking at the questions. Parts of the figure taken from [26].

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 found 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. We further propose ways to mitigate these language biases by creating subset of hard questions that require additional contextual cues to answer.

1. Introduction

Language has long been an integral part of visual understanding. From objects [4, 16] to human actions [12], categorization of visual data has lead to rapid developments in computer vision techniques, especially with deep learning. However, language is a much more powerful tool, and researchers recently have started to be apply it to domains beyond simple classification. To that end, various tasks such as image captioning [28] and Visual Question-answering (VQA) [1] have been proposed. VQA has arguably emerged as one of the most popular vision tasks, primarily due to its simple setup and clear evaluation.

MovieQA: QA tasks are particular 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. To that end, a recently released dataset, MovieQA [23], has extended 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. Similar ideas have been pursued in other works as well [14]. MovieQA dataset has 5 different categories for the QA task based on the following different types of story from which to do the QA task: 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 combination of visual and text data, whereas the remaining 4 are pure text based tasks.

While there has been a reasonably large amount of work in this direction, most methods [13, 17, 20, 27] 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 – which we name WikiWord embeddings – on unsupervised Wikipedia plots, so as to capture the narrative structure of movie plots. We find that this simple model outperforms *all* reported methods on MovieQA [23] 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 exploit such aligned data [3]. In contrast, we learn embeddings in an unsupervised fashion from *un*aligned movie plots $\{\text{plot}_i\}$, which are fine-tuned on training examples of $\{(\text{question,answer})_i\}$ pairs. 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 is primarily because visual information was not used for generating the QAs. Amazon Turkers generated the QAs just by reading the movie plots, without watching the movies. The movie clips were later programmatically aligned to movie plot lines and the questions. Another important thing we found is for a lot of QAs, words from movie plots were *copy-pasted* into the question and the correct answer choice, but not for the incorrect answer choices. Resulting into correct answer choice being very similar to movie plot lines and at the same time being very distinct from the incorrect answer choices. This makes it very easy to pick the correct answer by just looking at the question.

Fixing the bias: Inspired by efforts for uncovering and fixing the bias in visual QA [6], we demonstrate that our language model can be used to both expose language bias and build a stronger MovieQA benchmarks. We show, for various well-known QA models including ours, that the performance is much lower on our new benchmark, opening up avenues for further research on joint video-language understanding.

Why is this relevant for vision? Because our central technical contribution is a language model, one might argue that it is not relevant for a vision audience. While we do not propose a novel vision model, we propose novel baselines for widely-known vision benchmarks. It is crucial to ensure that strong baselines are introduced for the tasks at hand, to ensure meaningful progress is made. We feel that our results are very relevant for the MovieQA community and for future joint language-vision datasets. Importantly due to the presence of language bias in the dataset, the utilization of visual information by existing models for MovieQA can be questionable and the numeric results in terms of accuracy can give a false sense of working of these models.

The paper is organized as follows. Section 2 discusses related work. Section 3 introduces our WikiWord embedding model and proposes a modified benchmark protocol that reduces this bias. Section 4 provides empirical results on the MovieQA benchmark leader boards and results on new our benchmarks. Section 5 provides insights into what our simple QA only model is learning.

2. Related Work

Video and language: Joint learning of language and vision has been explored in various different ways. This includes movie descriptions [22], video understanding through fill in the blank [18], video retrieval [24], character co-referencing [21] and image captioning [28]. Lots of work have focused on using movies [8, 21, 24], because movies provide with time synchronized audio, subtitles and

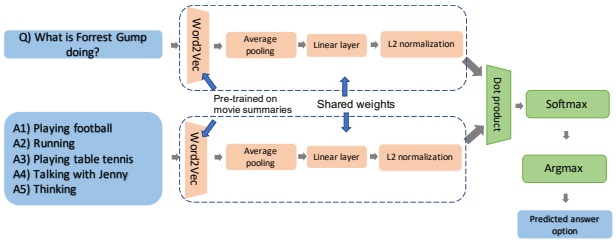

Figure 2: **WikiWord Embedding model.** It takes as input the question and 5 answer choices. For every word in the sentence (of question and answer choices) 300 dimensional word embedding is computed using word2vec. The word2vec is pretrained on movie plots and it's weights are kept fixed. These word level 300-d vectors are average pooled to get a 300-d sentence level vector which is then passed through a linear layer and is then L2 normalized. Dot product similarity is computed for the 300-d representation of question and the 5 answer choices, and the one with highest value is picked as the model's predicted answer option.

videos.

Visual QA task: Question answering provides an easy and unambiguous evaluation metric for joint language and vision tasks. The task is to predict correct answer from a list of options for a given question based on a story, which provides the context. Lots of 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 [23], constructed from movies, TVQA [14], constructed from TV series and TGIF QA [9], constructed from GIFs. Additionally, there has been work on reading comprehension [7], which are the purely language based QA datasets. We focus on MovieQA as one of the primary datasets for this task, consisting of movie clips, subtitles and questions from 140 movies. Some of the recent video models on this dataset include Attention to Attention Reasoning [17], Focal visual-text attention [15], Multimodal Dual Attention Memory [13], Layered Memory Network [27] and Read Write Memory Network [20].

Biases in Visual QA datasets: A well known problem with many visual QA datasets is the bias towards language cues. For instance, the original VQA 1.0 [1] dataset was heavily biased towards positive answers in a binary question setting. [29] proposed a simple baseline model exploiting such biases and obtained comparable performance to most of the complex approaches at the time. In VQA 2.0 [5], the authors tried to reduce the language bias in the dataset by augmenting the original dataset. Recently [11] compared different reading comprehension datasets to provide sensible baselines including comparison of models which predict the answer by just looking at the question alone or by just looking at the story alone. We extend these ideas to the video QA domain and show it suffers from such biases too.

3. Our Approach: WikiWord Embeddings

Classic formulations: Typical Question-Answering models are inspired by reading comprehension tasks from educational psychology [10]. These tasks can be formalized as triplets consisting of the reference passage (to be comprehended), a question, and the putative answers. Contemporary QA systems create a scoring function that iterate over all putative answers, conditioned on the question and reference passage, returning the highest-scoring answer.

WikiWord embedding model: Our model, shown in Figure 2, notably removes any reference story passage (subtitles and videos in the MovieQA dataset). Instead, it simply computes a score for each answer conditioned on the question, and returns the highest-scoring answer. Importantly, WikiWords makes use of word embeddings (word2vec [19]) pre-trained on a collection of Wikipedia movie plots, *including the Wikipedia plots of movies in the test set of question-answer pairs.* Important thing to mention is that the word embeddings are learnt from movie plots in an unsupervised way, without looking at the questions and ground-truth answers. Hence one is allowed to use movie plots from test set as well for training word embeddings according to the protocol from MovieQA authors. The default word embedding provided by MovieQA authors is also trained on movie plots in the test set.

The embedding layer takes as input the sentence (the raw question and all the answer choices) and for each word in the sentence computes a 300 dimensional vector representation using the pre-trained word2vec [19] model. The 300-d vector representation of every word in the sentence are then average pooled to get a 300-d vector representation of the whole sentence, which is then passed through a linear layer (initialized as an identity matrix) and then l2 normalized. The output dimensionality of the linear layer is again 300, as it is simply a linear projection of the average pooled word embeddings without any non-linearity. This is done for question and all 5 answer choices to get a 300-d vector and then we compute the dot product similarity of all 5 answer choices with the question, which is then passed through a softmax layer to get 5 probability values. Argmax over this gives the final predicted choice. The weights of the word embedding layer are kept fixed. The word embedding layer is shared for question and all the answer choices, as well as for subtitles used in our other experiments. For training we use cross entropy as the loss function. This architecture is based on the basic Visual Question-Answering framework provided in MovieQA paper [23].

Training data: WikiWords is trained in an unsupervised way on wiki-plot synopses that are movie summaries written by movie fans. They typically range from one to twenty short paragraphs. Interestingly, these plot synopses are used

to directly generate question and answers by crowd-sourced workers that do *not* have access to videos. This approach to question-answer construction might suggest one reason why plot-synopsis approaches (that have access to wiki-plots) dramatically outperform approaches that do not.

Default word2vec: At this point, it is useful to review the default word2vec provided by the MovieQA benchmark. The authors trained a similar word2vec as ours but use a larger set of Wikipedia movie plots. They use 1400 movies, which includes movies in train split, test split and the remaining movies are from outside the MovieQA dataset. We instead train our word2vec with movies present only in the MovieQA dataset (train and test splits). The purpose of word2vec is to provide general knowledge about the movies, in particular about the movie specific terms. In section 4.3 we show an ablation study on the amount of movie plots used to train word2vec. This provides good insight about our model.

3.1. Reducing the bias in MovieQA

In this section, we describe an approach that uses the previously described WikiWord embeddings to find and reduce the bias in the MovieQA dataset.

Background: Fixing the bias in VQA 1.0: Our approach is closely inspired by VQA 2.0 [5], which creates a 'balanced' dataset by augmenting the original 'unbalanced' dataset that contains a large degree of language bias. Because of the inherent language bias, the models were not using visual cues from images for a lot of questions. To rectify this, for a given question-answer-image triplet, the authors collected new images, which for the same question, have a different answer. They hypothesize that in this situation, models must make use of visual cues from the images in order to correctly answer both of the triplets. For example, They showed that existing state of the art VQA models performed worse on this newly balanced dataset. Further, they showed that existing models perform better when trained and tested on the balanced dataset, in comparison to when trained on unbalanced dataset and tested on the balanced dataset. This indicates that in the previous case, the models were exploiting the bias to obtain strong performance without learning from the visual information. With the balanced dataset, the models were forced to use visual cues.

We discovered similar language bias in the MovieQA dataset, which we believe makes it difficult to properly utilize videos. While dataset collection is a time consuming and expensive process, we propose two alternate quick fix approaches to mitigate this problem.

Approach 1: Easy-question removal: Our first approach towards fixing the bias is to remove the 'biased' QAs which can be answered by our simple QA only model (WikiWord embedding model). We refer to the remaining questions as the 'unbiased' QAs. In order to ensure that we don't overfit to the data when finding these biased questions from the training set, we use the predictions of our *untrained* QA-only model. That is we take our QA only model which uses the pretrained word2vec, and we do not further train it with question-ground truth answer pairs. This model obtains 40% test accuracy, and hence we can drop all the QAs it gets right. An important point to mention here is that in our model, the only trainable parameters is the linear layer which is initialized as an identity matrix. If it was randomly initialized, we likely would not achieve 40% test accuracy without training.

Our hypothesis is that these questions are the really easy and the most biased ones, that video based models can solve without needing context from videos. The remaining QAs (which we call 'unbiased' QAs) would be much harder and have higher possibility to need visual information in order to answer correctly.

Approach 2: Adversarial answer generation: Further we also tried another quick fix to *un-bias* the data. Now, instead of removing these 'biased QAs' we found from our previous approach, we augment them. That is, for a given 'biased' QA, we find the top 4 nearest neighbours in the word embedding space of the correct answer (from the list of all the answer choices across the different questions of the same movie). We then replace the 4 incorrect answer choices with these top 4 nearest neighbours, which are comparatively more similar to the correct answer choice.

We found that the original incorrect answer choices were very different from the correct answer choice in the word embedding space, making it easier for models to simply pick the correct answer choice without looking at the videos. Thus with this approach we create a new set of QAs which consists of 'unbiased' QAs plus 'biased' QAs which are 'augmented' by the nearest neighbours to make it harder to answer. We call this as the augmented dataset.

4. Experiments

4.1. Dataset

The purpose of MovieQA dataset is for building models which can understand stories both in videos and text through the task of question-answering. The dataset is built from texts and videos clips taken from commercial movies. The task here is to look at the story and given a question and 5 possible choices pick the correct one. The story can be in the form of text or video clips, based on this the dataset is divided into text based QA tasks and video based QA task. The full dataset consists of 14944 QAs taken from 408 movies. For the video based task in which movie clips and the corresponding movie subtitles form the story, there are 6462 QAs taken from 140 movies. This is the main focus of our work. For the text based tasks there are multiple kinds of text sources which form the story. This includes

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

Table 1: MovieQA Leader board for Subtitles category at the time of submission. Our model achieves about 5% higher accuracy then the second best submission.

Leader board submission	DVS
Our QA-only model	49.65%
MovieQA benchmark [23]	35.09%

Table 2: MovieQA Leader board for DVS category at the time of submission. Our model achieves about 15% higher accuracy then the second best submission.

Leader board submission	Scripts
Our QA-only model	45.49%
Read Write Memory Network [20]	39.36%

Table 3: MovieQA Leader board for Scripts category at the time of submission. Our model achieves about 6% higher accuracy then the second best submission.

subtitles, movie scripts, movie plots and DVS (Descriptive Video Service). Each of these have associated QAs. Subtitles and movie plots are available for all the movies and hence they can be used to answer all the QAs in the dataset.

4.2. Evaluation

Leader board results: The dataset is divided into train, validation (val) and test splits. The ground truth answers to QAs in test set are not released and one has to submit the predictions on the server for evaluation. Since the test server submissions are limited to once every 3 days, we follow the standard practice and do all our ablation experiments on the validation set. The train set is further divided into train (90%) and dev set (10%), the later is used for hyper-parameter tuning. All the splits are movie specific. Tables 1, 2, 3 and 4 show the results of our model on various categories on test server.

Table 5 shows the performance of different input modalities (QA only, subtitles, videos and videos+subtitles) for the top model on leaderboard with publicly released code. We make modifications to their code to do our experiments with different modalities and word embeddings and so some of the numbers don't exactly match with those in their paper. The accuracy after removing the video and subtitle inputs i.e. QA only version is very similar with video + subtitle version, indicating model is just exploiting bias and not actually utilizing videos or subtitles.

Leader board submission	Movie: Video+Subtitles
Our QA only model	46.98%
New method to optimize all MEM network (no details, anonymous submission)	45.31%
Multimodal dual attention memory [13]	41.41%

Table 4: MovieQA Leader board for Video+Subtitles category at the time of submission. Our model achieves about 1.5% higher accuracy then second best submission on leaderboard which is an anonymous one. Our model achieves about 5% higher accuracy then the second best published result.

Modality	Google	MovieQA	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 5: Validation experiments with different input modalities and word2vec on the model with publicly released code, Layered Memory Network [27]. It can be seen that using subtitle or videos doesn't help, they have similar accuracy as that when the model just uses questions to predict the answer. This shows the question-answer bias because of which the model is not able to utilize videos or subtitles. Also the table shows that in general the accuracy of the model (for all different modalities) increases as we use better and better word embedding from generic one like Google's to movie specific one like ours. This shows the importance of using the right word embeddings.

Better baseline for MovieQA: We show a very simple model which outperforms all the existing video based complex models which supposedly are utilizing videos, on 4 out of 5 categories by predicting answers by just looking at questions. Because of the language biases, the usefulness of videos by the models can be very questionable.

We noticed for most submitted visual QA models [13, 17, 20, 27] in MovieQA leaderboard, authors would conduct ablation study to compare the performance of their models for showing the relative usefulness of subtitle and video components (by individually switching them off), but not of their model which simply uses QA (by switching off both the video and subtitle components). We suggest (as we show in table 5) a good evaluation criteria when proposing a new video based QA model would be to check the performance of their model *after turning off both the video and subtitle inputs*. The accuracy of this simplified QA only model (which learns to predict answer by just looking at the question) would serve as the appropriate baseline for

Word2vec type	Movie plots for training w2v	Train accuracy (w/o training)	Train accuracy (stopping epoch)	Val accuracy (w/o training)	Val accuracy (stopping epoch)
MovieQA w2v [23]	General + train + val	27.70%	41.67%	26.74%	38.71%
Google w2v [19]	Google News	17.84%	30.40%	14.56%	20.31%
Ours	Val	20.30%	24.43%	40.51%	41.98%
Ours	Train	40.19%	57.46%	18.39%	19.30%
Ours	Train + val	39.90%	51.64%	38.48%	**49.88%**
Ours	General	21.34%	21.44%	17.17%	18.17%
Ours	General + val	21.31%	27.26%	34.76%	36.11%
Ours	General + train	36.77%	55.33%	16.59%	19.63%
Ours	General + train + val	36.01%	54.40%	32.73%	41.53%

Table 6: Experiments with QA only model (for movies+subtitle task) with different amount of movie plots used for training word2vec. This table shows the importance of different word embeddings. Generic word embedding like Google's (row 2) gives really poor accuracy. And using a good 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 to this movie plots, for 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.

their actual proposed model. The delta in the performance of the simplified QA only version and the full version of the same model would be a better evaluation criteria for the proposed full model. The newer datasets in this direction, like TVQA [14] provide this important baseline. Additionally training and evaluating models on the 'unbiased' QA subset created by our method can serve as a better evaluation measure for the video QA models.

4.3. Ablation Study - Importance of word embeddings

Movie specific words: Since questions and answers both are text, choosing the right word embedding is critical to exploit the bias. We experimented with word2vec trained on different data - 1) Google w2v (trained on 100 billion words from Google News dataset, has vocabulary of 3 million words) 2) MovieQA w2v (provided by the authors, which is trained on about 1400 movie plot synopses, this includes plots for all 408 movies in the MovieQA dataset) 3) Our different versions of w2v trained on different amount of movie plots.

Since the questions and answers are based on movies, they contain lots of movie specific vocabulary. It is really important for models to know movie characters and movie specific vocabulary to even understand the questions. For example, given a question containing the word 'Quidditch', the model should have some sense about what it is. That it's from the movie Harry Potter. That's where unsupervised word embeddings trained on appropriate data comes comes into picture.

Issues with generic word embeddings: If one uses a generic word embedding trained on huge datasets like the Google's w2v it may not have the movie specific words in it's vocabulary, and typically the out of vocabulary words are initialized as random vectors, due to which they totally loses the connection to the movie. Even if the movie related words are there in the vocabulary, all the movie related words of the same movie would most likely be grouped tightly in the word embedding space, which means the models won't be able to differentiate the fine grained importance of these words. The missing words in vocabulary of general word embeddings are mostly character names and other movie specific terms, hence if these words occur in question-answers the model would have very poor understanding of the questions themselves, forget the story. So it's important to train word embedding with movie related content especially for small datasets like MovieQA wherein the models cannot learn directly from the raw data. Good word embeddings hence can provide the models with general knowledge about the movie related entities which is essential for a small dataset. In figure 3 we compare the t-SNE [25] visualization of a google's word2vec (a generic embedding) and our WikiWord Embedding's word2vec. We observe the words related to a single movie cluster together in our embedding, while are jumbled in the generic one.

Google and MovieQA word2vec: Table 6 shows the performance of different pre-trained word2vecs when we are just using our simple QA only mode to predict the answers. Since the test set server submissions are limited to once in 3 days, we test and show results on the val set. Table 6 shows the accuracy of val and train sets, without training our QA only model (just using the pre-trained word2vec) and after training our simple QA model. As can be seen in the second row of the table 6, Google w2v

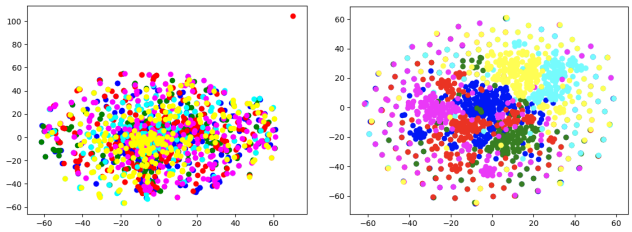

Figure 3: Left is t-SNE visualization of word embeddings based on google w2v and the right one is for our WikiWord Embedding w2v. We show them for words taken from 6 different movies, words from same movies have same color. Left plot shows for a generic word embedding like google w2v, words from different movies are all jumbled up together and hence they loose semantic meaning. Whereas for the right plot, words from same movie are all clustered together and away from those from other movies. Hence the words preserve movie specific semantic meaning, which is essential when questions and answers contain movie specific entities.

Type	Our model	TVQA baseline [14]
Original dataset	49.88%	32.50%
Only biased	99.41%	47.80%
Only unbiased	25.68%	22.50%
New augmented	31.82%	28.44%

Table 7: Comparison of performance on different splits of MovieQA dataset for 2 different QA only models. The first row shows the original dataset. The second row shows the subset of original dataset which is biased i.e. our QA only model is able to correctly answer them. The third row **(Approach 1: Easy-question removal)** is the subset which our QA only model is unable to correctly answer, resulting in chance level accuracy. These are the unbiased QAs and hence is the hardest split, and which would need information from videos and subtitles. Last row **(Approach 2: Adversarial question generation)** is our new proposed split which is combination of unbiased QAs and the biased QAs which are replaced by augmented QAs. This is comparatively harder split then the original one and is of the same size as the original split.

performs really poorly and gives close to chance accuracy (20%), on the val set after training. This is primarily because the movie specific words are missing in it's vocabulary. MovieQA's word2vec (provided by MovieQA authors) itself gives about 38.71% accuracy (first row) on the val set after training. The MovieQA word2vec is trained on about 1400 movie plots, majority of these are from movies not in the dataset.

Our word2vec: We now experiment with different subsets of the movie plots to train the word embedding. In table 6, 'val' refers to word2vec trained on movie plots of the movies in val set *(on which we test our model)*, 'train' on movies in train set, and 'general' on the 1400 movie plots minus those in train and val sets. We observe, that even without training, our simple QA model is able to get high accuracy (40.51%) on the val set (when we train the embedding on val set). Finally, training the word2vec on train+val sets leads to best performance on the val set (49.88%). Hence, just using plots which are part of the dataset leads to the best accuracy. Adding movie plots from outside ('general'), degrades the performance and we get results and trend similar to that of MovieQA.

4.4. Performance after reducing the bias

To further ensure that our unbiased dataset is competitive for multiple models, we show the performance of our QA-only model and that of the baseline QA-only model proposed in TVQA [14] dataset. The TVQA QA only baseline model is based on the context matching module followed by an LSTM layer. Unlike our QA only model it avoids average pooling of the embeddings of words to get sentence level embeddings. Table 7 compares performance when QA only models are trained and tested with the original dataset, only the biased subset, only the unbiased subset and new augmented dataset. We observe the fixed datasets are harder for both models, with the unbiased dataset (approach 1) being more difficult than the augmented one (approach 2). We believe the fixed datasets can benefit from models that exploit visual information.

5. Analysis

What does WikiWords learn?: In a way our simple QA model, the pre-trained word2vec model is trying to memorize the occurrence of nearby words in the movie plot synopsis and since the question-answers are made by AMT workers by only looking at the movie plot synopsis, it is able to correctly answer the QAs in half the dataset. Figure 4 shows the predictions of our simple model with 'train+val' word2vec which are correct and figure 5 shows the predictions which are incorrect. It also the highlights the prominent words in the question, the correct answer and the line in the movie plot form which the QA was made by the AMT workers.

We found that the model first tries to select the answer choice which has highest number of movie specific words as that in the question, this happens because in this case the word embedding of question and the selected answer would be very close. The another thing which model tries to do if the previous thing doesn't hold is to select the answer whose movie specific word(s) occur adjacent to the movie specific word(s) of the question in the movie plots (since in word2vec space nearby text words have very high dot product similarity). Again this ensures that the word embeddings of question and the selected answer choice would

Figure 4: **QAs which are correctly predicted by our simple QA only model and hence are the biased QAs.** Correct answer is highlighted in green. Light blue coloured words are the movie specific words common between the question and the line in movie plot from which the question was made by Amazon Mechanical Turkers. Dark blue underlined words are the movie specific words common between the correct answer and the line in movie plot. For example for the question in 2nd column, the model predicted A3) because 'Ping pong' (movie specific word in the correct answer) is the only word that appears close to 'Chinese teams' (movie specific word in the question) in the movie plot. For the incorrect answers options, the words in them don't occur in the movie plot and hence these options are are very different semantically in word embeddings. Due to this reason it's easy to find the correct answer without using any other information like from videos, hence making it a biased QA.

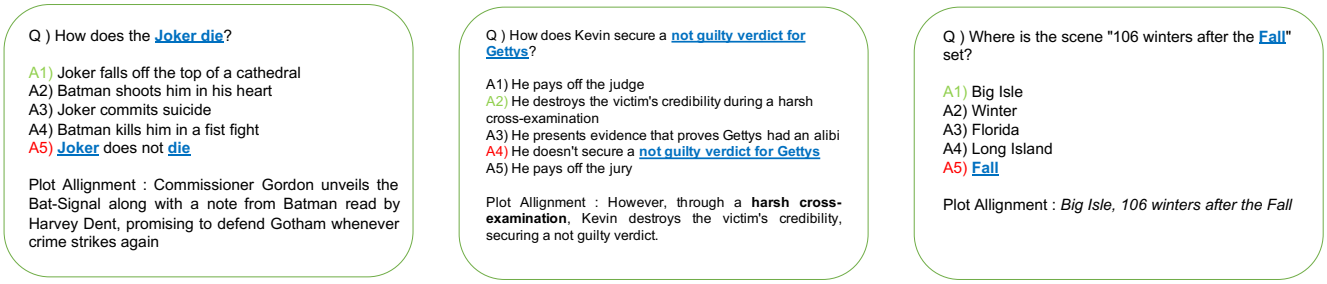

Figure 5: **QAs which are wrongly predicted by our simple QA only model.** These are the QAs which are more likely to be less biased. Prediction of the model is in red and the correct answer is in green. For example in 3rd question, the model predicted A5. This is because amongst all the answer choices the word 'Fall' in A5) is the only common movie specific word amongst the words in the question. Hence A5) would have very high dot product similarity in the word embedding space with the question and so the model predicted it as the answer.

have very high similarity. And surprisingly just doing this our simple model close to 50 percent accuracy on the video based QA task with only looking at question and picking the answer.

6. Conclusion

We show that the MovieQA dataset has language bias and present a simple QA only model that exploits it. We train it in unsupervised manner on movie plots and achieve state of the art performance on four of the five categories on the leaderboard at the time of submission. These language biases make it harder to analyze the effect of visual input in existing state of the art models. To mitigate this we propose a simple fix of removing the QAs which our simple QA-only model gets correct. We believe that this unbiased QAs could provide for a better evaluation metric for video based models.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: visual question answering. *CoRR*, abs/1505.00468, 2015.

[2] M. Azab, M. Wang, M. Smith, N. Kojima, J. Deng, and R. Mihalcea. Speaker Naming in Movies. *ArXiv e-prints*, Sept. 2018.

[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.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[5] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. *CoRR*, abs/1612.00837, 2016.

[6] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017.

[7] K. M. Hermann, T. Kociský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. *CoRR*, abs/1506.03340, 2015.

[8] Q. Huang, Y. Xiong, Y. Xiong, Y. Zhang, and D. Lin. From Trailers to Storylines: An Efficient Way to Learn from Movies. *ArXiv e-prints*, June 2018.

[9] Y. Jang, Y. Song, Y. Yu, Y. Kim, and G. Kim. TGIF-QA: toward spatio-temporal reasoning in visual question answering. *CoRR*, abs/1704.04497, 2017.

[10] M. A. Just and P. A. Carpenter. *The psychology of reading and language comprehension.* Allyn and Bacon, 1987.

[11] D. Kaushik and Z. C. Lipton. How much reading does reading comprehension require? a critical investigation of popular benchmarks. *CoRR*, abs/1808.04926, 2018.

[12] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[13] 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.

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

[15] J. Liang, L. Jiang, L. Cao, L.-J. Li, and A. G. Hauptmann. Focal visual-text attention for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6135–6143, 2018.

[16] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[17] C.-N. Liu, D.-J. Chen, H.-T. Chen, and T.-L. Liu. A2a: Attention to attention reasoning for movie question answering.

[18] T. Maharaj, N. Ballas, A. Rohrbach, A. C. Courville, and C. J. Pal. A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[19] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

[20] S. Na, S. Lee, J. Kim, and G. Kim. A read-write memory network for movie story understanding. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[21] A. Rohrbach, M. Rohrbach, S. Tang, S. J. Oh, and B. Schiele. Generating descriptions with grounded and co-referenced people. *CoRR*, abs/1704.01518, 2017.

[22] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, and B. Schiele. Movie description. *International Journal of Computer Vision*, 2017.

[23] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. MovieQA: Understanding Stories in Movies through Question-Answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

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

[25] L. Van Der Maaten and G. E. Hinton. Visualizing data using t-SNE. In *J. Mach. Learn. Research*, 2008.

[26] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555, 2014.

[27] B. Wang, Y. Xu, Y. Han, and R. Hong. Movie question answering: Remembering the textual cues for layered visual contents. In *AAAI*, 2018.

[28] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[29] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus. Simple baseline for visual question answering. *CoRR*, abs/1512.02167, 2015.