Who Framed Roger Rabbit?
Multiple Choice Questions Answering about Movie Plot

Daria Dzendzik
ADAPT Centre
School of Computing
Dublin City University
Dublin, Ireland
daria.dzendzik@adaptcentre.ie

Carl Vogel
School of Computer
Science and Statistics
Trinity College Dublin
the University of Dublin
Dublin, Ireland
vogel@tcd.ie

Qun Liu
ADAPT Centre
School of Computing
Dublin City University
Dublin, Ireland
qun.liu@adaptcentre.ie

Abstract

This paper introduces an approach to the task of multiple-choice question answering based on a combination of string similarities. The main idea of this work is to run a logistic regression over the concatenation of different similarity measures. Evaluating our model on the MovieQA plot data-set we obtain 79.76% accuracy, outperforming prior state-of-the-art results.[1]

1. Introduction

Question answering (QA) is a basic task in Natural Language Processing (NLP). The multiple-choice question answering (MCQA) is a sub-task of QA where several candidate answers are provided for each question.

In this paper, we describe our work with a multiple-choice question answering data-set for automatic story comprehension MovieQA [11]. The data contains almost 15,000 multiple choice question answers obtained from over 400 movies, Wikipedia plot synopses, subtitles, and scripts. Every question is accompanied by three to five answer candidates, with only one correct answer. The task is to select the correct answers based on an additional text. Every movie story is different and contains a unique context. The questions are disparate, for example (1)-(6). A table 1 contains examples of two questions with provided answer candidates.

(1) *Why does Octavius kidnap Mary Jane?*

(2) *What are Jack's attempts to save Rose after Titanic's sinking?*

(3) *Who kills Sirius?*

(4) *When was Boris captured?*

(5) *How does Marian feel about Robin's band?*

(6) *Does Batman manage to escape from the prison?*

We assume that answers to all the questions may be found in the movie plot related to the questions. We are interested in text understanding. This is why we focus on the plot synopses as a source of additional information.

Traditionally questions can be classified as factoid and non-factoid. Factoid questions ask *Who? What? When?* and assume that answer should be number, date or named entity and etc. A majority of research work in the QA area focus on such questions. The second class usually starts with words *Why?* and *How?* and require lengthier answers with explanation and reasoning. Each of these classes has their own ways of finding the answer. MovieQA contains question of both types, and that fact makes the task more challenging.

The main contribution of this paper is that we propose and validate an approach based on logistic regression over text similarities for MCQA that achieves a performance of 79.76% accuracy, which is much better than baseline on the MovieQA Plot data-set. Our result outperforms the state-of-the-art accuracy of 78.52%.

This paper is organized as follows: we describe the general idea of our approach in §2; the model and features are discussed in §3 the results of the experiments are presented in §4; some error analysis is described in §5; previous work on Movie related QA and answers re-ranking is addressed in §6; future work is outlined in §7; finally, our current conclusions are articulated in §8.

[1] According to the published results on leader board: http://movieqa.cs.toronto.edu/leaderboard/#table-plot – last verified September 2017

<table>
<tr><td colspan="2">Story (Title: ’71, 2014)</td></tr>
<tr><td colspan="2">Gary Hook, a new recruit to the British Army, takes leave of his much younger brother <b>Darren</b>. &lt; ... > Hook steps outside the pub just before an enormous explosion destroys the building. <b>Hook flees once more into the dark streets.</b> &lt; ... &gt;</td></tr>
<tr><td>Factoid question</td><td>Non-factoid question</td></tr>
<tr><td>What is the name of Hook’s younger brother?
• His name is Carl
• <b>His name is Darren</b>
• His name is Jimmy
• His name is David
• His name is Tom</td><td>How does Hook react to the explosion?
• He flees into the building next door
• He goes back into the pub to check for survivors and help the wounded
• He finds a payphone and calls the police
• <b>He flees into the street</b>
• He yells for help</td></tr>
</table>

Table 1. Examples of factoid and non-factoid questions and candidates of answers from the MovieQA data-set. Bold marks the relevant part of the plot synopsis and the correct answer.

2. Approach

Exploring the data we can conclude that questions and answers in the MovieQA data-set are similar to movie plot text. It is important to find the right sentence in the plot description which supports the correct answer. Our approach consists of 5 main parts: (1) Preprocessing, (2) Sentence Extraction, (3) Similarity calculation, (4) `Tf-IDf` Extension, (5) Logistic Regression. The pipeline of our system is presented on Figure 2.

2.1. Preprocessing

We use a minimum of text preprocessing. We are working with string similarities on word and character levels, so the text representation is important for our method. At this stage, we delete dots at the end of each answer if they exist.

2.2. Sentence Extraction

There are two different ways to extract relative information. The first one is based only on a sentence level logarithmic term frequency–inverse document frequency (`Tf-IDf`) similarity, as will be discuss in §2.4. The second one is based on a number of similarities. Using those similarities, we extract *k (k = 1,3,5)* sentences from a plot related to question. Subsequently, the extracted sentences are concatenated to one string. Duplicates sentences are ignored.

Given a set of sentences from a text T, question q and a set of answers A_q, we consider F – a set of features derived from a number of similarity methods. First, we extract relevant sentence S' from T by applying f, where $f \in F$ as in (7), and k is a number of sentences to be selected.

(7) $S'_f = \{s_1...s_k : max_k(f(q, s_i)), \forall i\ s_i \in T, \forall f \in F\}$,

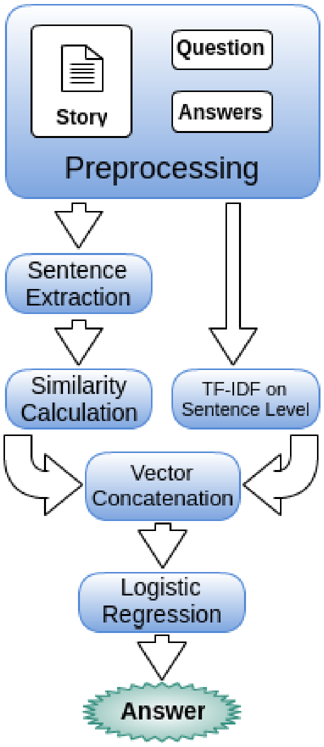

Figure 1. System pipeline

We concatenate all sentences from S' to one string S, as in (8), where $\forall j f_j \in F$, $p = |F|$ and k is fixed.

(8) $S_k = \cup_f S'_f = s_{1,f_1}+...+s_{k,f_1}+...+s_{1,f_p}+...+s_{k,f_p}$,

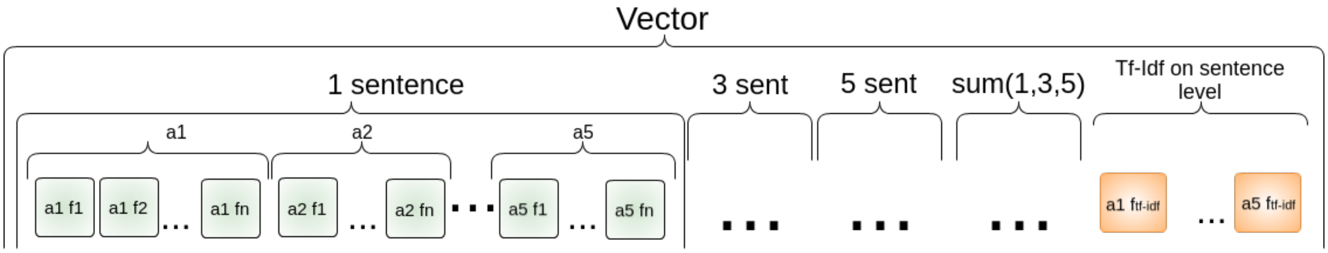

Figure 2. Feature vector concatenation, where n is the number of features, $n \geq p$

2.3. Similarity Calculations and Concatenation

Once we have a number of sentences selected for every question, we calculate similarity between the concatenation of sentences and every answer and between the concatenation of sentences and concatenation of question and answer.

Formally: for every answer a from A_q we calculate similarities between a and S, and between $q + a$ and S. So the set of similarity features V can be described as in (9) and $|V| = n$.

(9) $V_k(q, A_q, S_k) = \{f(a_i, S_k) \cup f(q + a_i, S_k) : \forall i \; a_i \in A_q, \forall f \; f \in F\}$

As was mentioned before, some similarities can be applied two times; that means that $n \geq p$.

Then, the similarities are concatenated as one vector. After that, we concatenate together vectors for different k and also we concatenate result with an element-wise sum of the vectors. From linear algebra point of view the sum makes sense as something between considering values.

2.4. TF-IDF Extension

This part of the system is inspired by results published on MovieQA leader-board[2] by University College London. For every question we extend the plot T with question q and answer candidates $a_j \in A_q$. We implement the natural term frequency `TF` and a logarithmic variant of `IDF`. Cosine similarity is calculated between every sentence in the plot and the question, and between every sentence and every answer candidate for the question. As shown in 10, the result is the maximum of sum this two similarities over all sentences and all answers.

(10) $Sim_{tfidf}(a_j, q, S) = max_i(cosine(t(q), t(s_i)) + cosine(t(a_j), t(s_i))), \forall j \; a_j \in A_q, \forall i \; s_i \in T$

where t is the `Tf-IDf` representation.

We extended our feature vector with logarithmic `Tf-IDf` similarities for every answer. The full concatenation process is presented om figure 2.

2.5. Logistic Regression

Above we describe the feature vector based on different text similarity measures. The vector contains the information about a question and all its answer candidates. In the final stage, we run a logistic regression which predicts the answer. To be more specific, we are trying to encode the most relevant part of the feature vector and obtain the answer.

3. Model and Features

Here, we describe the similarity features in detail.

3.1. Simple Similarities

We consider four main simple types of similarity:

Tf-idf - simple cosine similarity between `Tf-IDf` string representations.

Bag of words - a primitive bag of words measure shows the ratio of answer (or question + answer) words which exist in the sentences. See (11).

(11) $bow(a, s) = \frac{|w_a \cap w_s|}{|w_a|}$

In (11), w_a is bag of words from the answer (or the question + the answer), and w_s is bag of words from the chosen sentences.

Window slide - gets all possible sub-strings from a sentence selection via window slide. The window has a size equal to a length of the answer. This measure returns the highest ratio of sequence match between answer (or question + answer) and all sentences' sub-strings (see 12).

(12) $wSlide = max_i(\frac{2 * M_i}{T_i})$

In (12), T_i is the total number of elements in both sequences: the answer and i–sub-string, and M_i is the number of matches.

Character N-gram - this measure is very similar to `Window Slide` but this feature works on character level. The size of the window is limited by parameter N (We consider N = 2,3,4,5 characters). As a result, we get the ratio of N-gram overlap including white spaces between the answer (or the question + the answer) and the sentences.

[2] http://movieqa.cs.toronto.edu/leaderboard/#table-plot – last verified September 2017

3.2. Word2Vec features

Word2vec and **Word2vec baseline** - we also consider two measures based on word2vec representation. We used a pre-trained model from [11]. `Word2vec baseline` returns a score across all sentences in the plot, question, and answers, and `Word2vec` is a cosine similarity between word2vec representations of the answer (or question + answer) and the selected sentences.

3.3. Skipthought features

Skip-thoughts [6] is a model which is trained on the continuity of text from books and represents semantic and syntactic information. According to its authors [6, p.1], the model "... tries to reconstruct the surrounding sentences of an encoded passage. Sentences that share semantic and syntactic properties are thus mapped to similar vector representations."

To calculate skipthoughts-similarity we encode question, answers, story and selected sentences with the pre-trained model.

Skipthoughts cosine baseline returns the score for each answer. It is a sum of dot products between question and story, and the dot product between answers and story.

Skipthoughts cosine is a cosine similarity between the selected sentences and the answer (or the question and the answer).

4. Experiments

4.1. Data

MovieQA[3] data-set contains 14944 questions with up to 5 answers candidates and 3 types of additional text knowledge: wiki-plot synopses, subtitles, and scripts. As was mentioned before, MovieQA contains both factoid and non-factoid question (See examples in Table 1). Every question is annotated with the movie from Internet Movie Database[4] (IMDb). The data-set is split by authors by training, test and validation sets as shown on Table 2.

	Plot			Script		
	Train	**Val**	**Test**	**Train**	**Val**	**Test**
#Movies	269	56	83	133	26	40
#QA	9848	1958	3138	5236	976	1598

Table 2. Number of instances in the training, validation, and test sets of the MovieQA data-set.

Although in this work we are focused on wiki-plot data, we also tried our approach on movie scripts.

4.2. Experiment setup

Practically, for saving execution time, for sentence selection we use only limited number of similarities: `Tf-idf similarity`, `Window slide`, `Bag of words` and `Character N-gram` .

The parameters of the logistic regression are tuned on the validation set.

4.3. Results

We evaluate accuracy measure over MovieQA data-set and use plot synopsis and scripts as an additional text.

4.3.1 Movie Plot

Logistic Regression over Similarities			
Feature combination	**Train**	**Val**	**Test**
1(+)3(+)5(+)(1+3+5)*	76.66	74.8	76.04
+ Skth**	76.90	74.36	-
+ Skth + S-level tfidf	**80.39**	78.29	-
+ S-level tfidf	79.92	**78.39**	**79.76**

Table 3. Performance on training, validation and test sets. * - 1(+)3(+)5(+)(1+3+5) is concatenation of vector similarity for one extracted sentence, 3 extracted sentences and 5 extracted sentences. The last component is element-wise sum of the described vectors. ** - Skth is skipthought feature representation.

Table 3 contains results for logistic regression over different combinations of features. `1(+)3(+)5` is concatenation of vector similarity for one extracted sentence, three extracted sentences and five extracted sentences. `(1+3+5)` is element-wise sum of the described vectors. We concatenate all four components together to one vector: `1(+)3(+)5(+)(1+3+5)` (Table 3, line 1).

The addition of skipthoughts features (Table 3, line 2) improved result on train data for 0.23% but performance on validation set decline for 0.44%. Such fluctuation is not significant, so we can conclude that the skipthoughts representation similarity is not substantial for our method. As was described in §2.4 we extended the vector by adding `Tf-IDf` similarity on the sentence level for every answer option. Such combination showed the best outcome – 80.39% accuracy on the training set (Table 3, line 3), but the performance on the validation set is 78.29%. We ran our system with `Tf-IDf` on sentence level but without skipthought similarity and obtained the best result on validation set – 78.39% and on test set as well – 79.76% accuracy. This result outperforms the current state-of-the-art accuracy of 78.52%.

4.3.2 Movie Script

We also tried our approach on MovieQA script data (see Table 4). The combination of the extracted sentences performs

[3] http://movieqa.cs.toronto.edu/home/ – last verified September 2017

[4] http://www.imdb.com/ – last verified September 2017

Logistic Regression over Similarities			
Feature combination	**Train**	**Val**	**Test**
1(+)3(+)5(+)(1+3+5)*	27.14	26.53	-
S-level tfidf	28.61	27.76	-
1(+)3(+)5(+)(1+3+5) + S-level tfidf	35.31	30.32	24.16

Table 4. Performance on train, validation and test sets. * - 1(+)3(+)5(+)(1+3+5) is concatenation of vector similarity for one extracted sentence, 3 extracted sentences and 5 extracted sentences. The last component is element-wise sum of the described vectors.

with 27.14% accuracy on the training set and 26.53% on the validation set (Table 4, line 1). The `Tf-IDf` representation on sentence level shows the outcome – 28.61% and 27.76% on the training set and the validation set correspondingly (Table 4, line 2). The combination shows 35.31% accuracy on the training set, 30.32% on the validation set but only 24.16% on the test set (Table 4, line 3).

4.4. Feature performance

Additionally, we evaluate the accuracy of each feature performance separately on the plot data. Table 5 and Table 6 contain accuracy for each similarity measure on the training set and validation set, respectively.

The sentence level `Tf-IDf` similarity achieves the accuracy of 72.96% and 72.52% on the training set and the validation set correspondingly. The majority of string similarity features works in the range between 50-63% accuracy. Noticeably, semantic features perform in range 25-48% accuracy. The low output of skipthought features explains why excluding this representation improves the overall result of the model.

	1 sentence		3 sentence		5 sentence	
	AvsS	qAvsS	AvsS	qAvsS	AvsS	qAvsS
w2v bas	47.12		47.56		47.75	
w2v cos	41.50	45.78	30.08	28.55	28.06	25.64
skth bas	31.62		26.21		25.77	
skth cos	35.25	31.63	34.17	26.49	33.74	26.21
tfidf	58.56	60.66	53.72	56.15	50.69	52.30
overlap	60.10	60.93	60.48	62.65	58.98	61.56
wSlide	60.32	45.52	63.25	50.83	63.65	52.30
2gram	54.09	55.21	42.09	45.49	36.20	40.24
3gram	60.60	60.53	60.17	60.89	58.02	59.07
4gram	60.10	60.33	62.32	63.10	62.05	62.96
5gram	57.80	58.79	61.42	62.79	61.64	62.98
S-lvl tfidf	72.96					

Table 5. Separate performance of all features on the train data. AvsS is a similarity between answer and sentences. qAvsS is a similarity between a concatenation of question and answer and sentences.

	1 sentence		3 sentence		5 sentence	
	AvsS	qAvsS	AvsS	qAvsS	AvsS	qAvsS
w2v bas	47.75		47.34		47.24	
w2v cos	43.36	48.51	29.67	28.44	27.78	25.12
skth bas	32.17		25.94		26.71	
skth cos	34.98	32.99	33.65	27.68	33.75	26.40
tfidf	59.65	62.25	53.67	56.12	50.61	53.26
overlap	60.92	61.49	61.18	62.81	59.60	61.64
wSlide	61.64	45.14	64.19	52.14	63.89	53.21
2gram	55.00	56.12	42.39	46.37	36.51	40.04
3gram	62.46	63.17	60.77	62.20	57.55	59.95
4gram	61.95	62.05	64.35	64.65	63.99	64.65
5gram	59.95	60.62	62.61	63.78	62.76	64.19
S-lvl tfidf	72.52					

Table 6. Separate performance of all features on the validation data

The results of running logistic regression over only one selected sentences, only three selected sentences, only five selected sentences and the sum of them are presented in Table 7. Our observation is that value of accuracy increases with the increasing the number of extracted sentences. As was shown in §4.3.1, the best result is obtained by the combination of the features.

5. Error analysis

In this section, some errors on the plot data are discussed. Three main classes of errors can be highlighted. Note that in many cases these classes are overlapping. See table 8 for some examples.

The most common error (around 70% of mistakes) is caused by misunderstanding the context. This category includes rephrasing including synonyms and wrong references (line 1 and 2 in Table 8). Our method includes semantic representation (word2vec and skipthought). Apparently, as discussed in §4.4, our use of these features do not work satisfactorily.

The second big class of errors (about 32.5%) based on the fact that some questions request the information which spread along two or more sentences (line 2 and 3 in Table 8). The idea to select more than one sentence (three and five sentences as well) came from a desire to increase a probability of selecting the right sentence and also be able to analyze information across the sentences. We have to admit that sentence selection is working on a sentence level. Also, the order of sentences is ignored during the selection. The most relevant text comes first that sometimes can distract the considering information.

We can say that sentence extracting module works well. Only around 8% of questions were supported by sentences which do not contain correct answer information. That leads us to the previous problem.

	1 sentence		3 sentence		5 sentence		Sum	
	Train	Val	Train	Val	Train	Val	Train	Val
Log regr	60.47	61.64	72.02	71.34	73.87	72.36	75.35	74.36
+S lvl tf-idf	77.70	76.86	78.42	77.22	78.33	77.17	79.10	78.19

Table 7. Results of logistic regression run over separate combination features for one, tree, five extracted sentences and also the sum of them.

Question	Story sentence	Predicted answer	Correct answer	Explanation
What was Laura's dead husband profession?	Laura (Sharon Stone) works as a closet and drawer organizer and is **the widow of a race car driver**.	Closet and drawer organizer	Race Car Driver	Rephrasing : *widow of — dead husband profession*
Where does Walter kill Dietrichson?	**Walter Neff** (Fred MacMurray), < ... > After **Dietrichson** breaks his leg, **Phyllis drives him to the train station** for his trip to Palo Alto for a college reunion. **Neff** is hiding in the backseat and **kills Dietrichson** when Phyllis turns onto a deserted side street.	At Phyllis' house	On the ride to the train station	Information across the sentences. References: *Walter — Neff.* Rephrasing: *drives — on the ride*
What is the name of the leader of the Shopaholics Anonymous group?	Rebecca later returns home to renewed confrontations with her debt collector, so Suze makes her attend **Shopaholics Anonymous**. < ... > After one shopping spree she meets a friendly woman, **Miss Korch** (Wendie Malick), only to learn that she **is the group leader** and < ... > .	Suze	Miss Korch	Information across the sentences.

Table 8. Example of wrong answer selection. Bold marks relative part of the plot synopsis.

The described approach was designed for the plot understanding. We applied our method to the script texts and obtained reasonable results. More detailed performance error analysis is beyond of this paper.

6. Related work

6.1. MovieQA

The main results for MovieQA plot and script data-sets including this work are presented in Tables 9 and 10, respectively.

6.1.1 Plot

The baseline is introduced in [11]. The authors show four approaches for finding the correct answer: (1) *Hasty Student* (not in the table) chooses answers without looking to additional text. Best result was 28.14% accuracy on a question-answer similarity of the semantics of the sentence using SkipThoughts Vectors [6]. (2) *Searching Student (SS)* (not in the table) selects the answer based on a cosine similarity between different representations (TF-IDF, SkipThoughts, Word2Vec) of question-answers and corresponding additional data sources. (3) *Searching Student with Convolutional Brain (SSCB)* is a neural similarity model which considers the same representation and also combinations. Empirical evaluations show that SSCB is sensitive to initialization. The result of different runs of the system shows differences of up to 30% accuracy. Authors trained several networks using random start and picked the

System	**Train**	**Val**	**Test**
Tensor representation	-	-	78.52*
Convnet (tfidf + w2v)	-	-	77.63*
tfidf on sentence level	72.96	72.73	75.78*
CNN on word matching	-	72.1**	72.9**
SSCB tfidf + w2v	-	59.60	57.97*
SSCB Fusion	-	61.24	56.7***
MemN2N	-	40.45	38.43*
LogReg (sent selection + tfidf on sent level)	79.92	**78.39**	**79.76**

Table 9. The state-of-the-art results for the MovieQA plot data-set. * - results are obtained from the MovieQA Leader-board ** - results are obtained from [15] *** - results are obtained from [11]

model with the best performance on the internal validation set. This method achieves accuracy of 57.97% on plot synopsis data (Table 9, lines 5 and 6). (4) *MemN2N* is a memory network with additional embedding layer which encodes each multi-choice answer and uses an attention mechanism to find a relevant part of the story to the question. It achieves 38.43% accuracy on the test set (Table 9, line 7).

Others propose a four layer LSTM model [15] and investigate different comparison functions. This system achieved accuracy of 72.9% on the test set and 72.2% on the validation set (line 4) using combination of operations: SUBTRACTION, MULTIPLICATION, and NEURALNET(ReLU).

The sentence-level Tf-IDf similarity, which we discuss in section 2.4, was originally proposed by the Machine Reading Group[5] from University College London. This method shows result of 75.78% accuracy on test set (table 9 line 3). Another result from the same team is 77.63% on test set (Table 9, line 2). They use a SSCB method described in [11] with sentence level TF-IDF approach and word2vec representation. Unfortunately, no article is provided.

A team from National Taiwan University [6] achieved accuracy of 78.52% (Table 9, line 1). They represent paragraphs of the plot, a question and answer options as a tensor and use a sophisticated attention method to integrate them. A model consists of 3 layers: the first is a similarity mapping method, which computes the word embedding similarity between every word in paragraph and answer option (or question); the second is the attention based CNN matching; the third is a prediction layer which determines the final answer[7]. Unfortunately, this team also has not provided an article but some details can be found here: http://speech.ee.ntu.edu.tw/ tlkagk/MovieQA.pdf.[8]

The best result of 79.76% accuracy is obtained by our system (line 8). As we described, we use logistic regression over the vector of text similarities.

6.1.2 Script

In case of the script data the authors of the baseline [11] achieves 23.90% and 24.41% accuracy on the test set using *Searching Student (SS)* with Tf-IDf and word2vec repesentations correspondingly. The result of 37.05% was achieved by *MemN2N* with some modifications: a replacement the fully trainable architecture of the original MemN2N by word2vec embeddings and an addition a trainable, shared, linear projection layer which allows the memory network to answer using multiple choices.

System	**Train**	**Val**	**Test**
Read-Write-Memory-Network	-	-	39.36*
MemN2N	-	39.75	37.05*
SS** + w2v	24.43	25.72	24.41*
SS** + tfidf	21.21	20.90	23.90*
LogReg (sent selection + tfidf on sent level)	35.31	30.32	24.16

Table 10. The state-of-the-art results for the MovieQA script data-set. ** - SS is Searching Student from [11] * - results are obtained from the MovieQA Leader-board

The best result 39.36 is obtained by a collaboration of Vision & Learning Lab[9] from Seoul National University[10] and SK Telecom Video Tech. Lab[11]. According to leaderboard descriptions, an abstraction memory for a given story through Read-Write Network was created. It consists of Multi-layer Convolution, which collects neighboring memories to create a higher-level memory block. In addition, due to the long length of the MovieQA story, a sharpening operation is applied to the memory in order to perform the question and answers, so that more powerful attention is formed and the appropriate memory is retreated. There is no paper provided.

Our approach shows 24.16% accuracy on the test set. The result of our system outperforms the baseline on the train and validation sets but works not so well as the memory neural networks. On the test set, our system performs on the same level as the baseline.

6.2. General Question Answering and Movie Domain

This work draws inspiration from existing work on machine learning for question answering, more specifically, from non-factoid answer re-ranking. [1] use the Paragraph Vector model [7] to represent the question and the answer, then they concatenate these representations and use a fully-connected neural network to predict the score for the answer. This approach achieves state-of-the-art performance on a public data-set of how questions from Yahoo! Answers[12]. Another quite interesting approach on the same data-set was demonstrated by [5]. They use a discourse structure of sentences to improve the best answer selection. Later, [2] showed a hybrid mechanism of a neural network and handcrafted discourse features.

Apart from factoid and non-factoid, we can consider another type of QA categorization. Such data-sets like Yahoo!

[5] http://mr.cs.ucl.ac.uk/ – last verified September 2017

[6] http://www.ntu.edu.tw/english/index.html – last verified September 2017

[7] The description of the system is taken from the MovieQA leader-board.

[8] Last verified – October 2017.

[9] http://vision.snu.ac.kr/ – last verified September 2017

[10] http://www.useoul.edu/ – last verified September 2017

[11] https://www.facebook.com/skquantum/ – last verified September 2017

[12] https://answers.yahoo.com/ – last verified September 2017

Answers consist of a question and a set of user-generated answers without any further data. This is the answer re-ranking task: to put the community-selected answer to the top position. MovieQA also contains answer candidates and multi-choice question answering task can be considered as an answer re-ranking task, but there is a significant difference: availability of additional text for supporting the correct answer. Answers for movie data-set are normally short, while answers in Community Question Answering (like Yahoo! Answers) are quite long. On the another hand there are many QA systems based on reading comprehension task where usually no any answers or answers candidates are provided but additionally exist a small text passage where the answer can be found.

Apart from MovieQA, there are several others new QA reading comprehension data-sets were announced last year: Stanford Question Answering Data-set (SQuAD) [9], a Human Generated MAchine Reading COmprehension Data-set (MS MARCO) [8] and NewsQA [12]. Substantial interest in the task is evident.

Relatively recently researchers have started pay attention to domains like the movies in conducting text analysis. In 2011 [14] used Internet Movie Script Database (IMSDb)[13] corpus of films for learning models of character linguistic style. In the same year [3] realized Film Corpus 2.0. It contains scripts of 1068 from IMSDb. Also, there are 960 film scripts where the dialog in the film has been separated from the scene descriptions. One year later [13] introduced an annotated corpus of film dialogue for learning and characterizing character style. Last year [10] provides an insight of challenges for building QA systems, with a special focus on employing structured data. Authors inspected Wikipedia and DBpedia slices, including *Films*.

6.3. Text Similarity

The core of our method is a text similarity. We consider two main types of similarity (see §3): simple similarities (`Tf-IDf`, `WindowSlade`, `BagOfWords` and `N-grams`) and similarities of vector representation which we get using pretrained models (`Word2vec` and `Skipthought`). [4] considers more than 25 text similarities divided by five groups: character-based similarity, term-based similarity, corpus-based similarity, knowledge-based similarity, and hybrid similarity measures. Authors mentioned cosine similarity, `Tf-IDf` and `N-grams` similarities.

7. Discussion and Future Work

Now we briefly discuss our future work. We consider this paper as a work-in-progress and it mostly contains preliminary results. In the near future, we plan to make more detailed error analysis and extend the described model. We can use more distinct similarity metrics, for example, knowledge-based similarity from [4]. Also, there is still the open question of how to use semantic features properly to improve the model. Another possibility of development is an enhancement of the sentence selection module. For now, we focus on plot synopses analysis and only briefly tried our system on script data. In the future, we will adjust our approach for other subsets of MovieQA such as movie scripts and subtitles. Furthermore, we plan to explore the hybrid approach of machine learning and feature engineering to question answering task with additional text data.

8. Conclusions

We introduce a method based on text similarity and logistic regression for the answer selection task. Evaluating on the MovieQA plot data-set our method outperforms the state-of-the-art results of accuracy on the plot data. The feature performance evaluation and error analysis for plot data are provided.

We also tried our approach on MovieQA script data-set obtain a reasonable outcome.

Acknowledgments

This research is supported by Science Foundation Ireland in the ADAPT Centre for Digital Content Technology. The ADAPT Centre for Digital Content Technology is founded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund. We express special gratitude to Dasha Bogdanova and Henry Elder for discussions and support.

References

[1] D. Bogdanova and J. Foster. This is how we do it: Answer reranking for open-domain how questions with paragraph vectors and minimal feature engineering. In K. Knight, A. Nenkova, and O. Rambow, editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1290–1295. The Association for Computational Linguistics, 2016.

[2] D. Bogdanova, J. Foster, D. Dzendzik, and Q. Liu. If you can't beat them join them: Handcrafted features complement neural nets for non-factoid answer reranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 121–131, Valencia, Spain, April 2017. Association for Computational Linguistics.

[13] imsdb.com – last verified September 2017

[3] C. Danescu-Niculescu-Mizil and L. Lee. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*, 2011.

[4] W. H. Gomaa and A. A. Fahmy. Article: A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18, April 2013. Full text available.

[5] P. Jansen, M. Surdeanu, and P. Clark. *Discourse complements lexical semantics for non-factoid answer reranking*, volume 1, pages 977–986. Association for Computational Linguistics (ACL), 2014.

[6] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-thought vectors. *CoRR*, abs/1506.06726, 2015.

[7] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1188–1196. JMLR.org, 2014.

[8] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268, 2016.

[9] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016.

[10] S. Shekarpour, D. Lukovnikov, A. J. Kumar, K. M. Endris, K. Singh, H. Thakkar, and C. Lange. Question answering on linked data: Challenges and future directions. *CoRR*, abs/1601.03541, 2016.

[11] 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.

[12] A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman, and K. Suleman. Newsqa: A machine comprehension dataset. *CoRR*, abs/1611.09830, 2016.

[13] M. Walker, G. Lin, and J. Sawyer. An annotated corpus of film dialogue for learning and characterizing character style. In N. C. C. Chair), K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA), European Language Resources Association (ELRA).

[14] M. A. Walker, R. Grant, J. Sawyer, G. I. Lin, N. Wardrip-Fruin, and M. Buell. Perceived or not perceived: Film character models for expressive nlg. In M. Si, D. Thue, E. Andr, J. C. Lester, J. Tanenbaum, and V. Zammitto, editors, *ICIDS*, volume 7069 of *Lecture Notes in Computer Science*, pages 109–121. Springer, 2011.

[15] S. Wang and J. Jiang. A compare-aggregate model for matching text sequences. *CoRR*, abs/1611.01747, 2016.