Evaluating the Natural Language Understanding of a Machine by Answering Multiple Choice Questions for a Comprehension Text using proposed LKD Model

K.M.Arivu Chelvan, K.Lakshmi

Abstract*: *Machine Reading (MR) is an art of understanding text by the machine and one of the best tools to evaluate the understanding level of the machine is Reading Comprehension System (RCS) with Multiple Choice Questions (MCQ). In this paper, we proposed with a new knowledge representation, for understanding the given text, called Linguistic Knowledge Document (LKD). Such, LKD is generated from the given comprehension text. Natural Logic is used for generating the LKD. It is like an inference engine which contains all possible inference for each sentence in the comprehension text. The proposed LKD is acting like a human brain for the machine for answering the questions inquired by MCQA system. We use token based alignment model for finding answers from the LKD. We evaluate our system on RACE dataset and the obtained results are compared with recent methods. The comparison results show that the proposed model outperforms the recent results.

Keywords: Machine Reading; Reading Comprehension System; Multiple Choice Questions; Natural Language Inference; Alignment; Natural Logic

I. INTRODUCTION

The amount of information increases rapidly, through which it creates a major problem in finding the required information on need. The problem of finding useful piece of information requires more time for the human readers and also the understanding ability differs from one human to another. It motivates the machine reading research. The most important role of machine reading is to understand the given text. For the past two decades several methodologies had evolved for machine reading, such as pattern matching (bag-of-words), heuristic rules, machine learning approaches, textual entailment, semantic graph, natural language inference etc. Though there are various advantages and disadvantages in each approach; the main focus of this paper is to evaluate the Natural Language Inference (NLI) for machine reading.

Revised Manuscript Received on March 26, 2019

K.M.Arivu Chelvan, Research Scholar, CSE Periyar Maniammai Institute of Science & Technology Thanjavur, India.

K.Lakshmi, Professor, CSE, Periyar Maniammai Institute of Science & Technology Thanjavur, India.

Inference in conclusion is derived using some logical reasoning for the given text. NLI determines whether there is any reasonable inference found between a pair of sentences. Usually the pair refers to premise (p) and hypothesis (h) however it may change based on the application we are working on. The research on NLI over a decade found several approaches and are classified into shallow and deep approaches. Shallow approaches are based on lexical or semantic overlap, pattern-based relation extraction, or approximate matching of predicate-argument structure. Deep approaches depend on translation to FOL and theorem proving. Shallow approaches are robust and effective but it finds difficulty for inferences involving negation, quantifiers and antonym whereas deep approaches have power and accuracy but it finds difficulty in translating natural language to a formal language. In this work we combine shallow approach and deep approach with the help of Natural Logic to generate the inferences in a sentence.

The term Natural Logic was first introduced by Lakoff in the year 1970 and he defined natural logic as logic to categorize all valid inferences of a sentence in natural language. Natural Logic had a long history originated from Aristotle's Syllogism. It is a logical argument where the conclusion is derived from set of premises. Later Van Benthem & Sanchez Valencia [9] introduced monotonicity calculus. It is one of the most important mathematical features of natural language. It involves functional expressions to determine "upward" and "downward" inferences.

Natural Language Inference can be applied in several applications such as Question Answering, Semantic Search, Text Summarization, Machine Translation etc. The main motive of this paper is to develop a new method for NLI and to evaluate its performance over other methods. The best way to evaluate the machine reading is through answering multiple choices questions in a reading comprehension text.

In this paper, we proposed a new knowledge representation for the comprehension text called Linguistic Knowledge Document (LKD) for Multiple Choice Question Answering (MCQA) system. In this model the given comprehension text was generated as a knowledge representation using Natural Logic. The answer choices are represented as answer options as per the question. Using token based alignment LKD and answer options are aligned to find alignment score.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Based on the alignment score correct option is identified.

The paper is organized as follows: Section 2 gives an overview of MCQA systems. Section 3 describes Natural Logic. The proposed LKD is discussed in Section 4. Finally in section 5, MCQA system using LKD is evaluated and its results are compared.

II. Related Work

A. Reading Comprehension System

In general, the reading comprehension tests are conducted in schools for evaluating their student's language understanding level. The students are asked to read a comprehension and they need to answer few questions on the given comprehension. It is essential to notice two problems on the results produced by the students. First, the answers found by each student will differ based on the understanding level of the student. Second, the generated answers are further evaluated for finding better answers. The second case not only involves understanding but also involves language generation skills.

The first problem discussed above can be solved by Machine reading comprehension systems. We made an extensive survey on RCS and published [10]. Reading comprehension system was first proposed by L Hirschman et al. (1999). The proposed system is named as Deep Read. It accepts comprehension text as input and finds answers for the questions from the given text. It used pattern matching (bag of words) technique for finding the correct answers from the text. Later on several research works are carried out by different researchers in this area.

Ellen Riloff and Michael Thelen [4] had proposed a system called QUARC (QUestion Answering for Reading Comprehension). This system works under the principle of heuristic rules for all 'Wh' questions. Hwee Tou Ng et al. (2000) had proposed AQUAREAS (Automated Question Answering upon Reading Stories). It used machine learning techniques for finding answers. Ben Wellner et al. [5] proposed a system called ABCs (Abduction Based Comprehension system). This system used adductive inference engine to achieve results.

Juan Martinez-Romo and Lourdes Araujo [6] had developed a new system for question answering by constructing co-occurrence graph with words. Adrian Iftene et al., [24], Pinaki Bhaskar et al., [25] and Peter Clark et.al., [26] proposed their new ideas based on textual entailment in QA4MRE Challenge. Several research problems are proposed towards textual entailment in RTE challenge from 2006 onwards. Michael Hahn and Detmar Meurers [27] had proposed a system called CoSeC-DE for question answering. The approach behind CoSec-DE is semantic based approach.

The above mentioned reading comprehension test are based on answering 'Wh' questions. It requires separate mechanism for evaluating the generated answers. It causes problem while comparing the results of one system with another system. To resolve this issue Multiple Choice Question Answering (MCQA) systems were suggested for reading comprehension test. The best tool for evaluating the natural language understanding level of the machine was carried out using MCQA system.

Somnath Banerjee et al., [28] proposed a system based on Multiple Choice Question Answering system for entrance examination. They generated answer pattern for the question with its answer option. They used textual entailment approach for finding correct answer option. Xinjian Li et al., [29] had also proposed a MCQA system using RTE. Simon Ostermann et al., [30] presented a MCQA system for Entrance Exam and this system used alignment models for finding the answer. However the preprocessing on text was carried out by standard NLP tools. Helena et al., [31] presented a new approach where the given document and the answer options are transformed into graph based representation. Neil Dhruva et al. [32] presented an open domain reading comprehension system. It involved text similarity measures, textual entailment and coreference resolution for finding answers. Martin Gleize et al., [33] proposed a method to invalidate the answer options to find correct answer. It was done by generating Predicate Alignment Structure (PAS) to each answer options. Dominique Laurent et al., [34] had proposed a new method which used a special structure to save the results called CDS (Clause Description Structure). Yichong Xu et al [11] presented a neural model called Dynamic Fusion Network (DFN). This model used reinforcement learning to construct a network architecture to determine the attention vector and number of reasoning steps to proceed. Yi Tay et al [12] proposed a reasoning model called MRU (Multi-Range Reasoning Units) for implementing machine comprehension. This approach works under a 3 step process in the first step they construct learning representations for long and short-term context, next step finding relationship between intra document and finally encoding is done sequentially.

Though there are several works proposed for evaluating the understanding level of the machine but the research on this is wide open due to the scope for more improvement. In this paper we propose a new method for MCQA system. In this method, we are going to generate a knowledge representation named Linguistic Knowledge Document (LKD) using Natural Logic. The details about our approach are presented in the following sections.

B. Natural Logic

Natural Logic is a computational model for finding inference in natural language. Natural logic had a long history and there were several contributions by different authors. This part briefs such contributions. Inference using logics requires knowledge of both logician and linguist. First Order Logic (FOL) deals with predicates and quantification. Predicate plays an important role in FOL. It returns the truth value of discourse in a domain and requires formal representation.

Later this FOL is replaced by monotonicity reasoning proposed by van Benthem & Sanchez Valencia [9]. Here the predicates are replaced by predicates with smaller and larger extensions we call it as forward and reverse entailment. Monotonicity calculus proposed did not account on negation.

Narin et al., [35] proposed polarity propagation algorithm for negative sentences. Here the implications of a predicate in a sentence are embedded to form a complement construction about the sentence. They also worked on active verbs to find the truth value of negative sentences. It minimized the detection process on the entailment and contradiction.

Bill MacCartney and Christopher D. Manning [7] had proposed little advancement to monotonicity calculus. The proposed calculus had account for negation and exclusion. It has used five elementary entailment relations such as equivalent, forward, reverse, independent and exclusive.

Bill MacCartney [8] proposed a Natural Language Inference using Natural Logic where he uses 7 entailment relations also he proposed a phrase based aligner called MANLI aligner. The entailment relations are equivalent, forward, reverse, negation, alteration, cover and independence.

In this paper we use only first three operations such as equivalent, forward and reverse entailment of natural logic to construct our proposed knowledge representations.

C. Alignment

In NLP, the term alignment plays a vital role in different applications such as machine translation, natural language inference and question answering system. V. Jijkoun and M. de Rijke [22] proposed a Bag-of-words aligner. This aligner used frequency based term weighting for finding the similarity measures. Machine Translation (MT) aligners are GIZA++ & Cross-EM. GIZA++ was based on statistical machine translation using automation word alignment. Cross-EM was another important machine translation aligner based on unsupervised learning. It used symmetric word alignment between two simple asymmetric models to maximize a combination of data likelihood. Bill MacCartney et al., [21] introduced a NLI aligner called MANLI. The alignment used in MANLI was phrase-based alignment. It used lexical recourses and supervised learning for finding the results.

In this work we used a simple token based aligner as like bag of words. The procedure for finding the alignment score plays a key role in this approach. Here the alignment is done for each sentence in the LKD which helps to achieve better results. Findings of alignment scores are discussed in next section.

D. Dataset

Richardson et al., [14] proposed a dataset called MCTest. It uses fictional stories to answer multiple-choice reading comprehension questions. Hermann et al., [15] introduced CNN/Daily mail dataset. These datasets were constructed from news articles of CNN and Daily mail websites. F Hill et al., [16] presented a dataset called Children Book Test (CBT). The dataset CBT is constructed from freely books. Later BookTest dataset was introduced with more training examples by Ondrej Bajgar et al.,[17]. Rajpurkar et al., [18] present a new dataset for reading comprehension called SQuAD (Stanford Question Answering Dataset). It consist of over 1,00,000 questions from wekipedia articles. Adam Trischler et al., [19] introduced a new dataset called NewsQA. This dataset contains over 1,00,000 question-answer pairs generated by humans using CNN news articles. Payal Bajaj et al.,[20] introduced a large scale named MS MARCO (MAchine Reading Comprehension) dataset. The passages and questions are sampled from Bing web documents.

Guokun Lai et al., [23] introduced a new dataset called RACE for reading comprehension task. This dataset contains about 28,000 passages and about 1,00,000 questions formed by English instructors. RACE dataset comprised of RACE-M and RACE-H. RACE-M contains datasets of middle school whereas RACE-H contains datasets of high school. It covers a wide variety of topics for evaluating the ability in understanding and reasoning. We are using RACE dataset in this paper for evaluating our model.

III. Proposed Model

Our proposed model is consisting of two step process. In the first step the given comprehension text is generated into a knowledge representation called LKD. Second step is implementing a MCQA system using the generated LKD and the alignment score for finding the correct option for the question.

E. LKD Generation

Linguistic Knowledge Document (LKD) is a knowledge based constructed from the given comprehension text. We use natural logic for constructing the LKD. It involves four step process 1) Sentence Separation 2) Sentence Preprocessing, 3) Applying Natural Logic and 4) Constructing LKD. The block diagram of LKD is shown in Fig.1.

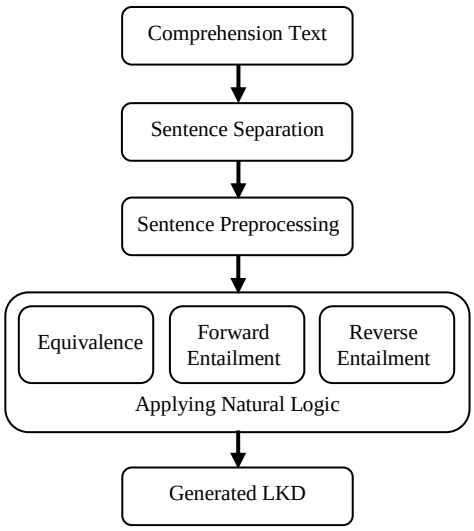

Fig. 1. Block Diagram of LKD

The first step in LKD construction is sentence separation. Several standard tools are there for doing this step. Stanford core NLP tool is used to separate all sentences from the comprehension text. The separated sentences are sequentially processed for constructing the LKD.

The separated sentences are preprocessed in this step. Preprocessing involves two major works one is Named Entity Recognization (NER) and another one is Part of Speech (POS) tagging. NER will recognize the pronouns used in the comprehension text. POS will tag each word of the sentence. Stanford core NLP tool is used to implement both NER and POS. The tagged words are transferred to the next step.

The most important step in LKD construction is applying Natural Logic to the sentences. We use three entailment relations in natural logic equivalence, forward entailment and reverse entailment. These relations are carried out at the lexical level of the sentences. Each word in the sentence is checked for the substitution. Equivalence relation is applied to the verbs in the sentence. It finds the

synonym of the word using word net tool. Assuming that if the synonym found for a verb contains *'V'* words then those *'V'* words form *'V'* sentences for the LKD. Entailment relations are applied to nouns in the sentence. A sentence may not contain a noun or it may contain one or more than one noun. We can apply either forward entailment or reverse entailment for the nouns in the sentence. Let us consider a sentence having *'N'* noun then it can be able to generate *'N'* new sentences. Hence, one sentence can form *'S'* sentences for the LKD. The term *'S'* defines multiples of V by N ie, S=VxN.

LKD contains sentences that generated after applying natural logic to each sentence in the comprehension text. LKD can be represented as sum of all sentences in the comprehension text after substituting equivalence and entailment relationship into the original sentence.

$$LKD = \sum_{S=i}^{n} V_i N_i$$

where, S is sentence in comprehension text, V_i is number of equivalence relation for i^{th} sentence, N_i is number of entailment relation for i^{th} sentence. The generated LKD is used for finding the correct answers for the MCQ that belongs to the comprehension text.

Algorithm 1: LKD Generation

```
Input: Comprehension Text as Document CTD
Output: LKD
NERD← NER(CTD)
S[ ]←NERD.Sentence Split()
N ← S.Length ()
for i =0 to N-1
sindex← 0
PTag[ ] ← POS(NS)
M ← PTag.Length()
for j = 0 to M-1
if (PTag[i].Split[1] == VN)
SYM[ ] ← Synonym(PTag[i].Split(0))
P ← SYM[ ].Length
For  k = 0 to P-1
KD[i][sindex++].Add(S[i].Replace(SYM[k]))
end for
else if (PTag[i].Split[1] == NN)
if(FENT ← Forward Entailment(PTag[i].Split(0))
                                        ==True)
LKD[i][sindex++].Add(S[i].Replace(FENT[k]))
for k = 0 to P-1
LKD[i][sindex++].Add(S[i].SYM[k].FENT)
end for
else
RENT ← Reverse Entailment(PTag [i].Split(0))
LKD[i][s index++].Add(S[i].Replace(RENT[k]))
end if
for k = 0 to P-1
LKD[i][s index++].Add(S[i].SYM[k].RENT)
end for
end if
end for
end for
```

F. MCQA system using LKD

The proposed LKD is used in MCQA system for finding the answer option. The system takes comprehension text and MCQ as input and produces an answer option through alignment model with an alignment score. Fig.2 shows the working flow model of MCQA system using LKD. The given comprehension text is converted into a knowledge representation called LKD, which is carried out using natural logic. The multiple choice questions with answer options are manually generated as answer sentences. The sentence in LKD and the answer sentence in the MCQ are considered as a pair for the alignment model. The alignment model used in our system is token based alignment model.

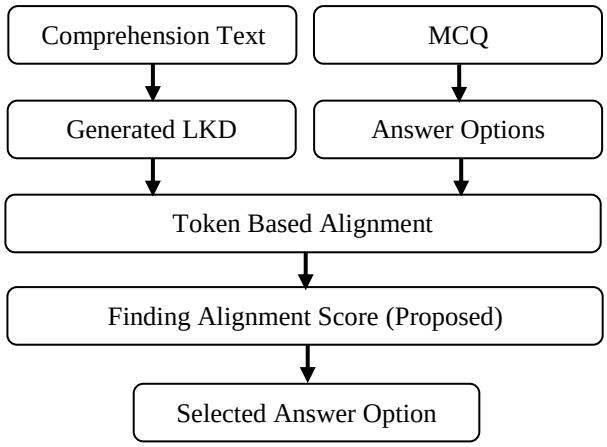

Fig.2. MCQA System using LKD

The source words from LKD and target words from answer sentence are mapped in a row column matrix to find the alignment score. LKD contains 'n' number of generated sentences for a single sentence in the comprehension text after applying natural logic. Every generated sentence in LKD will be mapped with each answer sentence in the aligner.

Every single sentence will get 'n' alignment scores for each answer sentence. A single question will have 'm' answer sentences (generated from answer options). Table 1 shows a representation of an alignment score, where the comprehension text contains 5 sentences and have 4 answer sentences.

Table 1: Alignment Score Matrix

	CT S1 (LKD 1- n1)	CT S2 (LKD 1- n2)	CT S3 (LKD 1- n3)	CT S4 (LKD 1- n4)	CT S5 (LKD 1- n5)
AS1	4	5	2	0	0
AS2	2	**7**	0	0	0
AS3	0	4	1	0	0
AS4	1	2	1	0	0

The above table contains a matrix where row represents Answer Sentences (AS) and column represents Comprehension Text (CT) and for each sentence in comprehension text there will be 'n' number of generated sentences in LKD. The value 'n' differs for each sentence in the text. Alignment score for each cell is computed using the following equation

$$AlignmentScore(AS1) = \sum_{LKD[1]=1}^{n} AScore$$

The following equation shows the way to find the alignment score for an answer sentence with respect to one single sentence of the comprehension.

In the same way the alignment score for each answer sentence in respect to each sentence in

comprehension text are calculated. The highest value in the alignment table is considered for choosing the correct answer option. In such way the highest value in the table is 7 and is associated with AS2 that concludes AS2 will be the correct answer option.

Algorithm 2 below shows the step by step procedure for implementing the MCQA system using LKD. The system is completely tested with RACE dataset and their results are discussed in the next section.

Algorithm 2: MCQA System using LKD

```
Input: L_K_D[][], Answer Sentence
Output: Correct Answer option
LKD[ ][ ]←L_K_D[][]
AS[ ]← Answer Sentence
for i = 0 to AS.Length-1
for j = 0 to LKD[ ].Length -1
for k = 0 to LKD[j] [ ].Length-1
TTokens[ ] = AS[i].SplitTokens()
STokens[ ] = LKD[j][k].SplitTokens()
AlignScore[i][j]+=Align(TTokens[ ], STokens[ ])
end for
end for
end for
Great ← 0
CorrectOption ← 0
for i = 0 to AS.Length-1
for j = 0 to LKD[ ].Length -1
if (AlignScore[i][j] > Great)
Great = AlignScore[i][j]
Correct Option = i
end if
end for
end for
return CorrectOption
```

IV. EVALUATION RESULTS

In this section, we discuss on the results achieved by our proposed MCQA system. The dataset used for the evaluation is recently released RACE dataset. The dataset contains MCQ collected from Chinese school English examinations. The dataset contains over 28,000 comprehension text and more than one l,00,000 multiple choice questions. The questions in the dataset contain wide variety and varying degrees of complexity. In our proposed system the articles are first generated into a knowledge representation called LKD. We then manually generate the choices into answer sentences corresponds to the questions.

RACE dataset is evaluated with human readers and it results about 95% accuracy. Our model achieves about 52% accuracy and it shows 5% improvement on comparing with Dynamic Fusion Network which produces 47% accuracy. Table 2 below shows the evaluation results.

Table 2: Performance comparison of proposed model

Model	RACE
Human Reader	95%
Dynamic Fusion Network	47%
MCQA System using LKD	52%

There is still plenty of scope for improvement in the Reading Comprehension research. We had concluded our work with few suggestions for future enhancement in our concluding section.

V. CONCLUSION

We focus on the evaluation of machine performance with respect to natural language understanding. This task is carried out using Reading Comprehension with Multiple Choice Questions. We used recent dataset called RACE to evaluate the understanding ability of our MCQA system. In this system we generated a knowledge representation called Linguistic Knowledge Document (LKD). We applied Natural Logic to generate the LKD. The generated LKD in turn helps in finding the alignment score of the basic alignment model called token based aligner. A single sentence of the article is generated in different forms of sentences using natural logic. Alignment score of each option with generated sentence are summed up to find the score for each option in the question. The option results with maximum alignment score is returned as answer for the question. The evaluation results are tested using RACE dataset. The comparison result shows there is almost 5% improvement over the recent models. One of the most important drawbacks of this model is computational complexity. It needs to align many generated sentences even though the alignment score of the system is zero. Another drawback is manually generated answer options with respect to the question. There is lot of scope for improvement in this model. We had considered only three operations in natural logic. We can apply another four operations to enhance the LKD. The polarity of the sentence can also monitored for improving the findings of alignment score. The scope on this research is wide open to resolve many unsolved problems.

REFERENCES

1. Hirschman L., Light M., Breck E. and Burger J.D., 1999. “Deep read: a reading comprehension system”, In: Proceedings of the 37th meeting of the Association for Computational Linguistics, Stroudsburg, PA, USA.
2. Penas A., Hovy E.H., Forner P., Rodrigo A., Sutcliffe R.F.E., Forascu C. and Sporleder C., 2001. “Overview of qa4mre at clef 2011: Question answering for machine reading evaluation”, CLEF (Notebook Papers/Labs/Workshop), pp. 1-20.
3. Anselmo Penas, Pamela Forner, Richard Sutcliffe, ˜Alvaro ´ Rodrigo, Corina Forascu, Inaki Alegria, Danilo Gi- ˜ampiccolo, Nicolas Moreau, and Petya Osenova, “Overview of ResPubliQA 2009: Question Answering Evaluation over European Legislation”, In Multilingual Information Access Evaluation I. Text Retrieval Experiments, CLEF 2009, Revised Selected Papers, volume 6241 of Lecture Notes in Computer Science, Springer, 2010.
4. Riloff E. and Thelon M., 2000. “A rule-based question answering system for reading comprehension tests”, In: Proceedings of the 2000 ANLP/NAACL Workshop on Reading comprehension tests as evaluation for computerbased language understanding sytems, Stroudsburg, PA, USA.
5. Wellner B., Ferro L., Greiff W. and Hirschman L., 2006. “Reading comprehension tests for computerbased understanding evaluation”, Nat. Lang. Eng., 12(4):305-334.
6. Juan Martinez-Romo and Lourdes Araujo, “Graph-based Word Clustering Applied to Question Answering and Reading Comprehension Tests”, CLEF 2011 Labs and Workshop - Notebook Papers. 19-22 September, Amsterdam, The Netherlands. Online Proceedings, 2011.
7. Bill MacCartney , Christopher D. Manning, Natural logic for textual inference, Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, June 28-29, 2007, Prague, Czech Republic.

8. MacCartney, Bill, and Christopher D. Manning. 2009. An extended model of natural logic. In Proceedings of the Eighth International Conference on Computational Semantics (IWCS-8).
9. van Benthem, Johan. 2008. A brief history of natural logic. Technical Report PP-2008-05, Institute for Logic, Language & Computation.
10. Arivuchelvan, K. M.; and Lakahmi, K.,2017. Reading comprehension system–a review. Indian J. Sci. Res, 2017, 14.1: 83-90
11. Xu, Y.; Liu, J.; Gao, J.; Shen, Y.; and Liu, X. 2017. Dynamic fusion networks for machine reading comprehension. arXiv preprint arXiv:1711.04964.
12. Y. Tay, L. A. Tuan, and S. C. Hui. Multi-range reasoning for machine comprehension. arXiv preprint arXiv:1803.09074, 2018
13. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. arXiv preprint arXiv:1704.04683 .
14. Matthew Richardson, Christopher JC Burges, and Erin Renshaw. MCTest: A challenge dataset for the open-domain machine comprehension of text. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2013.
15. Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In Advances in Neural Information Processing Systems, pages 1684–1692.
16. Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The Goldilocks principle: Reading children's books with explicit memory representations. In Proceedings of the International Conference on Learning Representations, 2016.
17. Ondrej Bajgar, Rudolf Kadlec, and Jan Kleindienst. Embracing data abundance: Booktest dataset for reading comprehension. arXiv preprint arXiv:1610.00956, 2016
18. Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2016.
19. Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. arXiv preprint arXiv:1611.09830 .
20. Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. arXiv preprint arXiv:1611.09268.
21. Bill MacCartney, Michel Galley, and Christopher D Manning. 2008. A phrase-based alignment model for natural language inference. In Proceedings of EMNLP.
22. Jijkoun and M. de Rijke. 2005. Recognizing textual entailment using lexical similarity. In Proceedings of the PASCAL Challenges Workshop on Recognizing Textual Entailment, pages 73–76.
23. G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. arXiv preprint arXiv:1704.04683 .
24. Iftene A, Gˆınsca A-L, Moruz M.A, Trandabat D, Husarciuc M, Boros E, "Enhancing a Question Answering System with Textual Entailment for Machine Reading Evaluation", CLEF 2012 (Online Working Notes/Labs/Workshop), 2012.
25. Bhaskar P, Pakray P, Banerjee S, Banerjee S, Bandyopadhyay S, Gelbukh A, "Question answering system for QA4MRE@CLEF 2012", In: CLEF 2012 Workshop on Question Answering For Machine Reading Evaluation (QA4MRE).
26. Peter Clark, Phil Harrison and Xuchen Yao, "An Entailment-Based Approach to the QA4MRE Challenge", Proc. CLEF 2012 (Conference and Labs of the Evaluation Forum) - QA4MRE Lab 2012.
27. Detmar Meurers, Ramon Ziai, Niels Ott, and Janina Kopp, "Evaluating Answers to Reading Comprehension Questions in Context: Results for German and the Role of Information Structure", In Proceedings of the TextInfer 2011 Workshop on Textual Entailment, EMNLP 2011.
28. Banerjee, S., Bhaskar, P., Pakray, P., Bandyopadhyay, S., Gelbukh, A, Multiple Choice Question (MCQ) Answering System for Entrance Examination, Question Answering System for QA4MRE@CLEF 2013. CLEF 2013 Evaluation Labs and Workshop Online Working Notes, ISBN 978-88-904810-5-5, ISSN 2038-4963, Valencia -Spain, 23 -26, 2013.
29. Li, X., Ran, T., Nguyen, N.L.T., Miyao, Y., Aizawa, A, "Question Answering System for Entrance Exams in QA4MRE", CLEF 2013 Evaluation Labs and Workshop Online Working Notes, ISBN 978-88-904810-5-5, ISSN 2038-4963, Valencia -Spain, 23 -26, 2013.
30. Simon Ostermann, Nikolina Koleva, Alexis Palmer and Andrea Horbach, "CSGS: Adapting a short answer scoring system for multiple-choice reading comprehension exercises", CLEF 2014 Working Notes, Sheffield, 2014.
31. Helena Gómez-Adorno, Grigori Sidorov, David Pinto and Alexander Gelbukh, "Graph Based Approach for the Question Answering Task Based on Entrance Exams", CLEF 2014 Working Notes, Sheffield, 2014.
32. Neil Dhruva, Oliver Ferschke and Iryna Gurevych, "Solving Open-Domain Multiple Choice Questions with Textual Entailment and Text Similarity Measures", CLEF 2014 Working Notes, Sheffield, 2014.
33. Martin Gleize, Anne-Laure Ligozat and Brigitte Grau, "LIMSI-CNRS@CLEF 2014: Invalidating Answers for Multiple Choice Question Answering", CLEF 2014 Working Notes, Sheffield, 2014.
34. Dominique Laurent, Baptiste Chardon, Sophie Negre, Camille Pradel and Patrick Seguela, "Reading Comprehension at Entrance Exams 2015", CLEF 2015 Working Notes, Toulouse, 2015.
35. R. Nairn, C. Condoravdi & L. Karttunen, 'Computing Relative Polarity for Textual Inference', Palo Alto Research Center, in Proceedings Inference in Computational Semantics, 2006.