CS657A: Information Retreival

Project Report

Automated Multiple Choice Questions Generator

Group 16

Abhinav Dudeja
21111001
abhinavd21@iitk.ac.in

Malkeet Singh Dhalla
21111035
malkeetsd21@iitk.ac.in

Manish Kumar
21111037
manishk21@iitk.ac.in

Manthan Kojage
21111039
manthank21@iitk.ac.in

Tabish Ahmad
21111060
tabisha21@iitk.ac.in

Indian Institute of Technology, Kanpur

Instructor
Dr. Arnab Bhattarcharya

Abstract

Objective question are becoming popular due to ease of evluatuon. Nevertheless, setting questions has always been time consuming. In the MCQs, their structure motivates an automated generation program. This project aims to build an MCQ generator using well known NLP techniques. We devised a workflow for this task which is able to generate relevant questions for the given text. We show that objective type questions have a good scope for automated generation system.

1 Introduction

Setting the question papers is a very time consuming task which can take several hours even for a test consisting of a few questions. If this process is automated it can save a lot of time for the teachers and institutes. To solve this problem we have made a MCQ generator which can automatically generate multiple choice questions for any given document.

Our system implements a pipeline which generates questions based on the context of the text document provided as an input. The workflow consists of pronoun replacement, relevant sentence and keyword extraction, question generation and wrong answers(distractors) selection.

2 Proposed Idea

Considering a human's perspective, we follow a similar method of framing the questions. A person frames questions by finding relevant parts of the text and produces similar enough options. We have proposed a architecture that uses various algorithms and NLP models to produce useful multiple choice questions from a document. We have developed a pipeline which gets the useful information out of the document and then, generate questions from it.

3 Methodology

3.1 Pipeline

We cannot just get the questions from any text, we first need to make it interpret-able enough for the machine to actually create grammatically correct and relevant questions. We have generated a pipeline to achieve this task. The pipeline has the following phases:

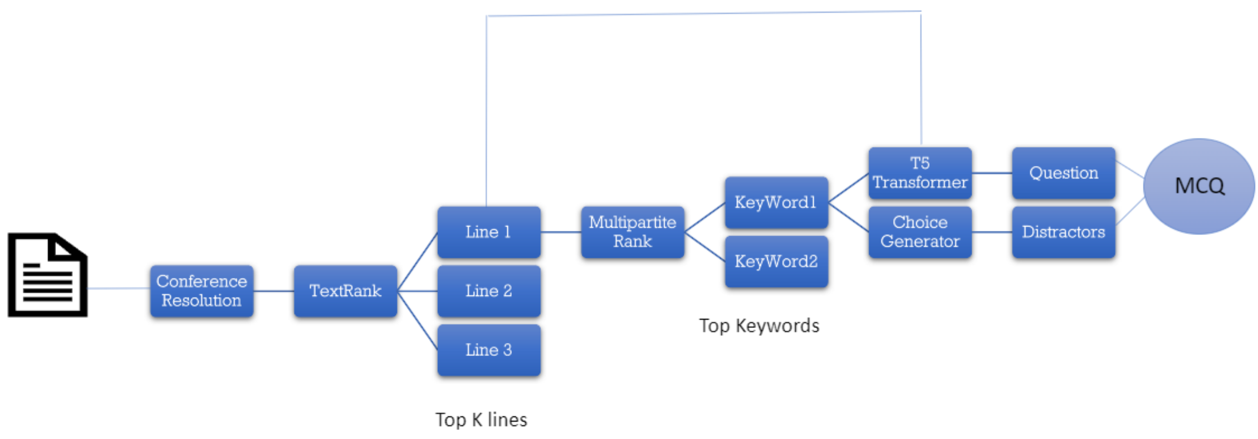

Figure 1: Pipeline

1. First, we convert all the pronouns by the nouns they point to. This helps in better understanding of what document is actually referring to in each line.
2. Now, we get the top-k sentences from the document based on their importance and relevance to what document is about.

3. After getting the relevant statement we extract keywords from these lines to help machine know around what the question is to be framed.
4. Questions are now generated using the context provided by the top-k lines selected in step 2 and the keywords extracted in step 3.
5. Finally, to complete the MCQ we generate meaningful distractors, according to the meaning the keywords hold in the statements from which they are derived.

3.2 Models Used

3.2.1 AllenNLP Coreference Resolution(1)

Coreference resolution is the task of finding all expressions that refer to the same entity in a text. Aim is to replace the pronouns in the sentence with the entity it represents. AllenNLP does this by finding clusters from position of words. If some word positions fall in same cluster, that means those words represent same entity.

3.2.2 TextRank(2)

TextRank is an extractive and unsupervised text summarization technique. Model uses sentence vector similarity graph to rank the sentences extracted from the document. Sentence with high similarity score are ranked higher. We have used this model to retrieve Top-K sentences.

3.2.3 Multipartite Rank(3)

Multipartite ranking is also a graph similarity model that is used for keyphrase extraction. The model doesn't only rely on the importance of the phrase in the document, position of the phrase is also taken into consideration, while extracting keyphrases.

3.2.4 BERT for WSD(4) & WordNet(5)

This model is used to generate distractors by word sense disambiguation. BERT uses the combination of the sentence from which the keyphrase was extracted and its meanings present in the WordNet dataset to return the most likely context of that keyphrase for that particular sentence.

WordNet is network of english words in which congnitive synonyms are clustered together. Its interlinking of specific senses of words makes it a useful tool for NLP based tasks. In this case, it gives words related to the correct answer, which can act as distractors for MCQs. Words are stored in hierarchical order and we have to find words under same umbrella terms(i.e. cohyponyms).

3.2.5 ConceptNet(6)

ConceptNet is a freely-available semantic network, designed to help computers understand the meanings of words that people use. It is more diverse that WordNet as it is based on Wikitionary, which gives it information regarding synonyms and antonyms.

ConceptNet also uses the hierarchical order to store the words and we have to find the cohyponyms similarly as in WordNet.

3.2.6 Sense2Vec(7)

Sense2Vec is a specialized word2vec model trained on reddit comments. There were no models trained with WSD hence, we have used this for distractors as a last resort. The embeddings of this model are more diverse than both the previous networks.

3.2.7 T5 Transformer(8)

T5 stands for text-to-text transfer transformer. T5 is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks and for which each task is converted into a text-to-text format. T5 works well on a variety of tasks out-of-the-box by prepending a different prefix to the input corresponding to each task, e.g., for translation, for question answering, for summarizing documents.

Model was trained on SQuAD(Stanford Question Answering Dataset)(9). The input is a (context, answer) pair and its output is questions.

4 Experiments

4.1 Summarization

We tried abstractive summarization using pegasus-xsum(10) to get meaningful sentences from the text, as it can help in generation of good quality questions. But, abstractive summarizers generated very short summaries which could not be used for question generation.

To overcome this problem we tried extractive summarization, which generates summary using statements from the given text as they are in the text. This method did generate summaries with more sentences but lack of meaningfulness resulted in poor quality of keyword extraction and question geneartion.

We finalized using AllenNLP Coreference Resolution along with TextRank to get enough lines of summary with meaningful information. Substituting pronouns with the refered nouns and then extracting top-k relevant lines gives a better base for question generation.

4.2 Keyword Extraction

We started with extracting keywords from complete text and then using the sentences which contain those keywords. With this method the question that were generated were not able to generalize and mostly unsatisfactory.

To get better quality keywords and questions, we then tried summarizing the text and extracting keywords from a smaller block of text. This helped in better generalization of the questions. Models such as rake, yake and multipartite were tried and best questions and keywords seem to be generated using multipartite rank.

4.3 Generating Distractors

We initially used BERT For WSD with WordNet, to get the distractors as choices for our MCQs. WordNet is based on American Lingua Franca, which makes it context limited. But, it gives good results in specific domains as it can be used with word sense disambiguation. To cover other domains, we then used ConceptNet which is more diverse. This helps in generalising our model. ConceptNet was not used with WSD as no models have been trained and thus, have been used only for domains not present in WordNet. Finally, we have used Sense2Vec to further expand the domain as it has been trained on a divesified dataset of reddit comments.

4.4 Question Generation

We tried t5 model for question generation process, as it is a text-to-text transformer model. Its results were relevant and meaningful as it generated grammatically and contextually correct questions.

5 Limitations & Future Work

5.1 Limitations

- Some of the outputs do not resemble a relevant question.
- Failed to generate distractors for some domains.
- Keyword selection can be improved.
- Distractors generated are highly dataset dependent.
- Distractors are not context dependent.

5.2 Future Work

There are some ideas we plan to apply in future:

- Expanding the variety of questions by including True/False, Fill in the blanks, Match the following.
- Other methods for generating distractors.
- NER to get context specific distractors.

6 Conclusion

We successfully developed Q/A system which extracts relevant keywords and generates meaningful questions. It reinforces the idea that structure of objective questions is befitting to automated question generation process. With customization for specific use-cases the model can be deployed in various environments like private institutions, competitive examinations, schools.

References

[1] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. Liu, M. Peters, M. Schmitz, and L. Zettlemoyer, "Allennlp: A deep semantic natural language processing platform," 2018.

[2] R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, (Barcelona, Spain), pp. 404–411, Association for Computational Linguistics, July 2004.

[3] F. Boudin, "Unsupervised keyphrase extraction with multipartite graphs," 2018.

[4] J. Du, F. Qi, and M. Sun, "Using bert for word sense disambiguation," 2019.

[5] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, p. 39–41, nov 1995.

[6] H. Liu and P. Singh, "Conceptnet — a practical commonsense reasoning tool-kit," *BT Technology Journal*, vol. 22, p. 211–226, oct 2004.

[7] A. Trask, P. Michalak, and J. Liu, "sense2vec - a fast and accurate method for word sense disambiguation in neural word embeddings," 2015.

[8] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2019.

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

[10] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," 2019.

[11] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

[12] R. Goutham, "Questgen.ai," 2020.