Automating reading comprehension by generating question and answer pair

Asking relevant and intelligent questions has always been an integral part of human learning, as it can help assess user understanding of a piece of text (a comprehension, an article, etc.). However, forming questions manually has always been an arduous task. Automated question generation (QG) systems can help alleviate this problem by learning to generate questions and answers on a large scale and in lesser time. Such a system has many applications in a myriad of other areas such as FAQ generation, intelligent tutoring systems, and virtual assistants.

Given a piece of text as a sequence of words, our goal is to generate syntactically correct, meaningful and natural questions along with answers to those questions.

Our approach to generating question-answer pairs from text is a two-stage process: in the first stage we select the most relevant and appropriate candidate answer, i.e., the *pivotal answer*, using an answer selection module, and in the second stage we encode the answer span in the sentence and use a sequence to sequence model with a rich set of linguistic features to generate questions for the pivotal answer.

Our sentence encoder transforms the input sentence into a list of fixed-length continuous vector word representation, each input symbol being represented

Question Generation: Illustration

Evaluated on Biogas Tutorials for training Village Masons

Sentence:

The reason is LPG burners are specifically used for high compression LPG only.



Demo: http://qg-system.herokuapp.com/

as a vector. The question decoder takes in the output from the sentence encoder and produces one symbol at a time and stops at the EOS (end of sentence) marker. To focus on certain important words while generating questions (decoding) we use a global attention mechanism. The attention module is connected to both the sentence encoder as well as the question decoder, thus allowing the question decoder to focus on appropriate segments of the sentence while generating the next word of the question. We include linguistic features for words so that the model can learn more generalised syntactic transformations.

We have used this question system for generating questions for improved reading comprehension as well as self-assesment by the user for several tutorials from http://spoken-tutorial.org/ meant for skill development and training. Below, we present our system diagram which can be tested at http://qg-system.herokuapp.com/



Prof. Ganesh Ramakrishnan, Department of Computer Science and Engineering, ganesh@cse.iitb.ac.in