Question Generation

Amit Kumar & Prashant Chaturvedi

August 31, 2021

Motivation

- We often read online article so this project is useful for testing concepts based on the recently read online articles.
- Question Answering on SQuAD challenge
- It uses natural language processing (NLP) techniques language modeling, part-of-speech tagging, named entity recognition, parsing, etc.

Introduction

- Problem : How can we automatically process some given text and generate applicable questions?

Solution ???

Fundamental NLP Tools

- Text Prepossessing

  - Removing noise(numbers, alphanumeric, punctuation and stopwords)
- Keyword extraction

  - Tokenization

  - Part of speech Tagging
- Key Libraries used

  - Rake-nltk

  - Word2Vec

  - spatial

  - networkx

Text preprocessing - Regex

- Regular expression is a language of different symbols and syntax that can be used to search for a piece of string within a larger string.
- It can be used in almost any coding language, and is very useful when trying to search for general string patterns.
- r‘A*’ matches A 0 or more times (‘’, ‘A’, ‘AA’, ‘AAA’, ‘AAAA’, etc.)
- r‘A+’ matches A 1 or more times (‘A’, ‘AA’, ‘AAA’, ‘AAAA’, etc.)
- r‘[a-z]*’ matches any lowercase letter 0 or more times (‘’, ‘ajrk’, ‘bor’, ‘q’, etc.)

Text preprocessing - Stopwords

- A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.
- We would not want these words to take up space in our database, or taking up valuable processing time.
- NLTK(Natural Language Toolkit) in python has a list of stopwords stored in 16 different languages.

Text preprocessing - Tokenization

- Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms. Each of these smaller units are called tokens.
- “This is a cat.” - [‘This’, ‘is’, ‘a’, cat’].
- We use the sent_tokenize() method to split a document or paragraph into sentences

Top Sentences - Word2Vec

- Word2Vec produces a vector space, typically of several hundred dimensions, with each unique word in the corpus such that words that share common contexts in the corpus are located close to one another in the space.
- finding words that are semantically similar to a word

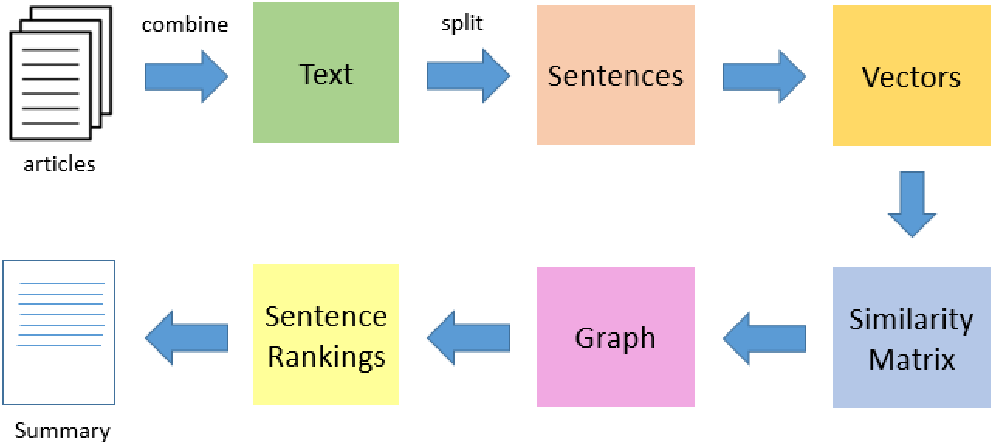
Top-Sentences - Padding

- Bits or characters that fill up unused portions of a data structure, such as a field, packet or frame.
- Typically, padding is done at the end of the structure to fill it up with data, with the padding usually consisting of 1 bits, blank characters or null characters.
- Padding sentence embedding using 0s to max_len. This has been done so as to maintain the same size embedding for each sentence.

Top-Sentences - Similarity matrix

- Initialize similarity_matrix of dimension N x N where N is the total number of sentences in the text
- Using 1-spatial.distance.cosine(), calculate the similarity between every two pairs of sentences.
- Convert the similarity matrix to a network/graph.

Top Sentences- TextRank



How is TextRank associated with Pagerank?

- Text sentences are used in place of WebPages
- The similarity matrix for index [A, B] is filled with similarity values between sentences A B rather than 1/(total links) from Page B to A which can be calculated using either cosine distance, maximum common words, etc.

Textranking Algorithm

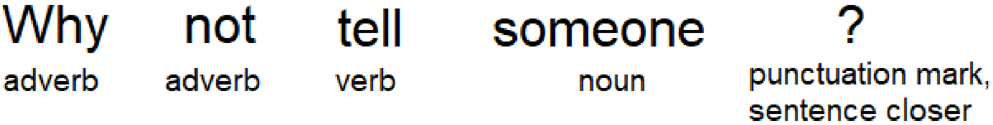
- Construct the similarity matrix between sentences
- Identify relations that connect such text units, and use these relations to draw edges between vertices in the graph. Edges can be directed or undirected, weighted or unweighted.
- Use Pagerank to score the sentences in graph
- Sort vertices based on their final score. Use the values attached to each vertex for ranking/selection decisions.

Fill in the Blanks- Rake-nltk(Rapid Automatic Keyword Extraction)

- Rake is a keyword extraction algorithm that is extremely efficient which operates on individual documents
- Rake is based on the observations that keywords frequently contain multiple words with with minimum lexical meaning.

Question formation - POS tagging

- Part-of-speech tags describe the characteristic structure of lexical terms within a sentence or text, therefore, we can use them for making assumptions about semantics



Why	not	tell	someone	?
adverb	adverb	verb	noun	punctuation mark, sentence closer

Question formation - Subject, Object, Predicate

- To generate question from the sentence we need to break up the sentence into subject, object and predicate using the above POS tags.
- Object is the verb/verbs occurring in the sentence. A Noun can be the Subject whereas a Predicate can be either a Noun or Adjective.
- Then apply NER (Named Entity Recognition) on the subject present in the triplet. In the above sentence the subject "He" will come under named entity "person".
- Depending on the type of named entity extracted for the subject, we can replace it with a list of wh-word. Replace person with the wh-word "who".
- "The aliens were killers" the triplet will contain, Subject→The aliens, Object→were, Predicate→killers which can be converted into "Who were killers" by replacing the subject with the wh-word depending on the Named entity extracted for the subject.

Conclusion

- We are able generate two types of question through given text which is Fill in the blanks and Wh's question.
- The techniques like language modeling, part-of-speech tagging, named entity recognition, parsing, etc. are directly applicable in question generation.
- There is no open source resource to identify subject, predicate, object from sentence.
- Due to the advances in online learning, automatic question generation and assessment are becoming popular in the intelligent education system.
- This program should critically analyzed the methods of objective question generation, subjective question generation with the learner's response evaluation.

References

- Mihalcea, Rada, and Paul Tarau. "Textrank: Bringing order into text." Proceedings of the 2004 conference on empirical methods in natural language processing. 2004.
- Rajpurkar, Pranav, et al. "Squad: 100,000+ questions for machine comprehension of text." arXiv preprint arXiv:1606.05250 (2016).
- Duan, Nan, et al. "Question generation for question answering." Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017.
- Heilman, Michael, and Noah A. Smith. "Good question! statistical ranking for question generation." Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. 2010.
- Smith, Noah A., Michael Heilman, and Rebecca Hwa. "Question generation as a competitive undergraduate course project." Proceedings of the NSF Workshop on the Question Generation Shared Task and Evaluation Challenge. 2008.
- Manning, Christopher D., et al. "The Stanford CoreNLP natural language processing toolkit." Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations. 2014.

The End