Geração de Perguntas Através de Mineração de Texto e Aprendizagem de Máquina

João Victor Galdino, Maely Souza Coutinho, Matheus Felipe

Mineração de texto, aprendizagem de máquina – Universidade Federal Rural de Pernambuco (UFRPE)
Recife – PE – Brasil

Departamento de Ciência da Computação – UFRPE

Abstract. *In this paper we seek to illustrate our process of research, exploration, training, testing and evaluation of three algorithms for the automatic generation of questions from a supporting text.*

Resumo. *Neste artigo buscamos ilustrar nosso processo de pesquisa, exploração, treino, teste e avaliação de três algoritmos para a geração automática de perguntas partindo de um texto de apoio.*

1. Introdução

A geração automática de questões é um assunto muito interessante para a área da educação, ele pode se traduzir em diversos outros problemas como geração de respostas certas, geração de respostas erradas, questões de verdadeiro ou falso, etc. Com o objetivo de conhecer mais sobre o assunto e fazer nossas próprias experimentações que escolhemos o tópico específico de geração de perguntas para abordar, portanto nossos resultados serão baseados nesse quesito, porém nesse paper também iremos mencionar a geração de respostas certas e erradas que chegamos a estudar com menos foco.

2. Estudando Artigo

No início do nosso trabalho fizemos uma leitura detalhada do artigo proposto pelo professor (A Systematic Review of Automatic Question Generation for Educational Purposes) e discutimos entre nós o que retiramos de mais importante da leitura. Após isso destacamos no documento essas partes mais críticas para o nosso entendimento e implementação.

3. Analisando Bando de Dados

3.1 Banco de dados de ciência

Depois de estudarmos o artigo iniciamos a análise do nosso primeiro banco de dados[2]. Como podemos perceber lendo a descrição do banco disponível no site ele possui os campos:

Question: Campo contendo a questão

Distractors: Campo contendo as opções incorretas de respostas

Correct Answer: Campo contendo a opção correta de resposta

Support: Campo contendo um texto que ajuda no entendimento da resposta correta.

O Banco está no formato .json e já possui uma divisão de treino e teste. As perguntas são na área de ciências e vem desse paper.

3.2 Banco de dados SQuAD

Após um período de teste e avaliação manual do primeiro banco, observamos que as respostas estavam bastante divergentes das perguntas na maior parte das vezes, dado que não conseguimos bons resultados nos nossos experimentos na área de ciências, como mencionado, seguimos o conselho do nosso orientador e optamos por trocar nosso banco para o mencionado pelo desenvolvedor do código que usamos como base. Segue o link onde está hospedado o novo banco: https://rajpurkar.github.io/SQuAD-explorer/

Como podemos perceber avaliando o banco ele possui diversos campos mas aqui destacamos os que usaremos:

Question: A pergunta

Context: Um texto que passa um contexto para a pergunta

Answer: A resposta correta

Diversas perguntas diferentes têm o mesmo contexto e não possuímos respostas erradas nesse banco. Os dados vêm em formato de .json e também já vem dividido em teste e o que ele chama de dev.

3.3 Um pouco sobre o SQuAD

O Stanford Question Answering Dataset (SQuAD) é um conjunto de dados de compreensão de leitura, consistindo de perguntas feitas por voluntários em um conjunto de artigos da Wikipedia, onde a resposta para cada pergunta é um segmento de texto, ou intervalo, da passagem de leitura correspondente ou a pergunta pode ser irrespondível.

3.4 Adaptando o dataset

Tivemos que fazer diversas mudanças na estrutura do banco pois ele veio em formato de dicionário com diversas chaves e alguns valores que não nos traziam informações relevantes. Após essa limpeza criamos o banco com 3 colunas com os campos que mencionamos acima.

4. Iniciando Testes

4.1. Palavras mais frequentes

Para retirarmos mais informações do banco decidimos fazer alguns testes preliminares. O primeiro que fizemos foi o de verificar se as palavras mais frequentes nos textos de suporte apareciam na resposta correta. Os resultados foram positivos com uma média de

0.866 calculada através da seguinte lógica: Era contada cada palavra da resposta correta que aparecia no suporte e dividimos essa soma pela quantidade total de palavras na resposta correta.

4.2. Similaridade

Já no estudo da similaridade entre os campos do banco de dados obtivemos resultados bastante diversos, nos dando a ideia de que não existe um padrão. Analisamos a similaridade de cada coluna com o texto suporte e também analisamos a similaridade entre a pergunta e as respostas, tanto corretas quanto erradas, e as respostas entre si.

Adicionamos ambos os resultados em colunas do nosso dataframe para possível utilização futura.

5. Pré-Processamento

Iniciamos o pré-processamento seguindo as indicações do artigo. Estamos na fase de pré-processamento básico de nlp, aplicamos Sentence splitting, Tokenization, POS tagging. Decidimos não utilizar o Entity recognition (NER) no momento, pois foi mencionado como uma possibilidade dependendo do banco de dados e do caminho para a geração de perguntas que vamos escolher. Inicialmente julgamos ser pouco influente nos resultados dado que o nosso banco possui questões voltadas para a ciência, citando muito esporadicamente entidades.

6. Código

Após o pré-processamento passamos nosso texto de suporte/contexto em uma função para achar e salvar uma das 10 palavras mais frequentes do texto através do nltk FreqDist e selecionamos essa palavra como a mais importante para guiar a geração de perguntas. Após isso passamos o texto com a palavra destacada para o algoritmo que usa as seções de geração de distractors e retirada de ambiguidade no processo.

6.1 T5

T5 (Text to text transfer) é um transformer based architecture que usa uma abordagem de texto para texto. Cada tarefa - incluindo tradução, resposta a perguntas e classificação - é lançada como alimento para o modelo de texto como entrada e treinando-o para gerar algum texto de saída.

Utilizamos esse algoritmo informando para ele o texto de suporte com a palavra-chave indicada e através disso conseguimos a geração das perguntas.

6.2 Retirando ambiguidade

Usamos o Word Sense Disambiguation (WSD) conseguimos obter um valor numérico que indica a probabilidade de cada uma das definições de ser a real dada um contexto, sendo assim a de maior valor é a definição que se enquadra na ocasião. Usamos isso para impedir que nossa palavra-chave seja interpretada errada e afete a próxima sessão da geração de distractors.

6.3 Geração de distractors

Para gerarmos os distractors ou respostas erradas usamos a palavra selecionada como a mais importante e usamos o Wordnet, um banco contendo diversas palavras e suas relações. Cada palavra tem sua palavra raiz (rosa tem como raiz cor) e várias palavras têm a mesma palavra raiz (verde também tem cor como raiz), sendo assim pegamos a palavra raiz da nossa palavra chave e as palavras relacionadas a palavra raiz que não são nossa chave serão nossos distractors.

6.4 BART

Nós também usamos o BART como outra forma de gerar perguntas de forma automática, para este algoritmo só precisamos informar o texto de suporte e ele é capaz de gerar uma pergunta que pode ser de resposta aberta ou fechada.

Aqui a seleção da palavra-chave é feita internamente, portanto apenas informamos o texto de apoio.

7. Um pouco sobre o SQuAD

O Stanford Question Answering Dataset (SQuAD) é um conjunto de dados de compreensão de leitura, consistindo de perguntas feitas por voluntários em um conjunto de artigos da Wikipedia, onde a resposta para cada pergunta é um segmento de texto, ou intervalo, da passagem de leitura correspondente ou a pergunta pode ser irrespondível.

8. Adaptando o dataset

Tivemos que fazer diversas mudanças na estrutura do banco pois ele veio em formato de dicionário com diversas chaves e alguns valores que não nos traziam informações relevantes. Após essa limpeza criamos o banco com 3 colunas com os campos que mencionamos acima.

9. Avaliação

Para a parte da avaliação buscamos no artigo base algumas ideias, a grande maioria dos estudos, porém, usam avaliação humana para definir se o conteúdo gerado é de boa qualidade, dado seus próprios conceitos. O método mais usado (21 estudos) foi avaliação de especialistas nos assuntos das questões avaliando se a questão era adequada ou não para a aplicação em uma turma de alunos. O segundo maior (15 estudos) foi comparação de questões geradas pela IA com questões geradas por humanos, inclusive foi o método utilizado no estudo do primeiro dataset trabalhamos antes (SciQ).

Outros métodos incluem usar estudantes para avaliar e responder as questões, para saber sobre dificuldade e usabilidade. Alguns desses métodos usaram métricas de text summarization para comparar a similaridade. Contudo, a maioria dos estudos não provê muitas informações sobre suas avaliações, como métodos, ou quantas pessoas fizeram parte da validação, ou se foi um processo remunerado ou voluntário.

Portanto, fizemos uma busca sobre possíveis métodos além da validação manual que poderíamos usar no nosso caso e escolhemos usar text comparison, usando a similaridade do cosseno dos embeddings, para comparar as questões e o contexto.

10. Resultados

10.1 T5

T5 tem uma média de aproximadamente 53% de similaridade entre as questões geradas e seus textos de apoio

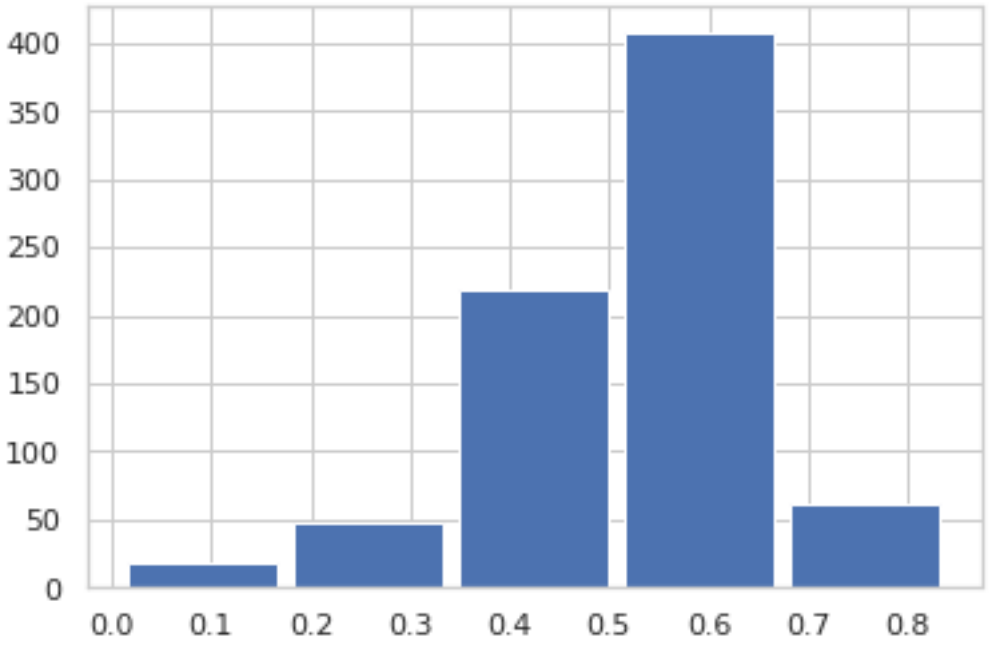

Figura 1: Visualização da avaliação do T5

10.1 T5 banco de dados de ciência

Como título de comparação fizemos uma avaliação de uma sample do modelo de ciências, surpreendentemente obtivemos melhores resultados do que nossa avaliação no novo banco de dados, chegamos a 66%, porém nossa amostra foi reduzida e comparamos apenas as perguntas.

11. Referências

1. Kurdi Ghader, Leo Jared, Parsia Bijan, Sattler Uli, Al-Emari Salam “A Systematic Review of Automatic Question Generation for Educational Purposes”
2. https://allenai.org/data/sciq
3. Welbl Johannes, F. Liu Nelson, Gardner Matt “Crowdsourcing Multiple Choice Science Questions”

https://www.semanticscholar.org/paper/Crowdsourcing-Multiple-Choice-Science-Questions-Welbl-Liu/932a5de79d8a8ebb75ea0c43493450fd9922e738

4. https://rajpurkar.github.io/SQuAD-explorer/

5. Rajpurkar Pranav, Zhang Jian, Lopyrev Konstantin, Liang Percy "SQuAD: 100,000+ Questions for Machine Comprehension of Text"

6. Rajpurkar Pranav, Jia Robin, Liang Percy "Know What You Don't Know: Unanswerable Questions for SQuAD"

7. https://www.sbert.net/docs/usage/semantic_textual_similarity.html

8. https://github.com/BPYap/BERT-WSD

9. Mike Lewis*, Yinhan Liu*, Naman Goyal*, Marjan Ghazvininejad,

Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, Luke Zettlemoyer "BART: Denoising Sequence-to-Sequence Pre-training for Natural

Language Generation, Translation, and Comprehension"