Leitura do artigo

Dense Passage Retrieval for Open-Domain Question Answering

(Karpukhin et al)

Conceitos importantes

"Open-domain question answering (QA)":

Responder questões usando uma base grande de documentos

Normalmente resolvido com um estágio de TF-IDF/BM25

Representação densa:

Tokens são representados por um vetor com *n* features. Esses vetores são gerados após um treinamento e a ideia é que vetores que representam palavras similares tenham algum grau de similaridade.

Contribuições do artigo

Sugestão para resolver o problema de QA usando representação densa, com os embeddings extraídos após o treinamento de 2 encoders usando uma base de treinamento relativamente pequena.

Contribuições do artigo

Treinamento:

O corpus é formado por um conjunto de passagens: $C = \{p_1, p_2, ..., p_M\}$

Cada passagem é formada por uma sequência de tokens: $p_i = [w^i_1, ..., w^i_{|pi|}]$

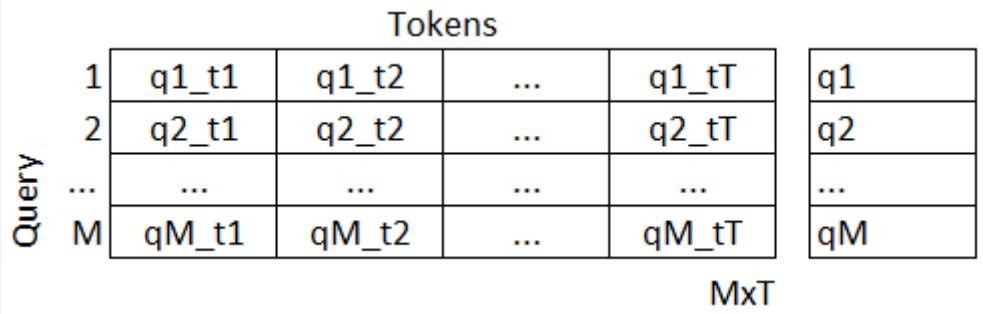
Há uma query associada a cada uma dessas passagens: (q_i, p_i)

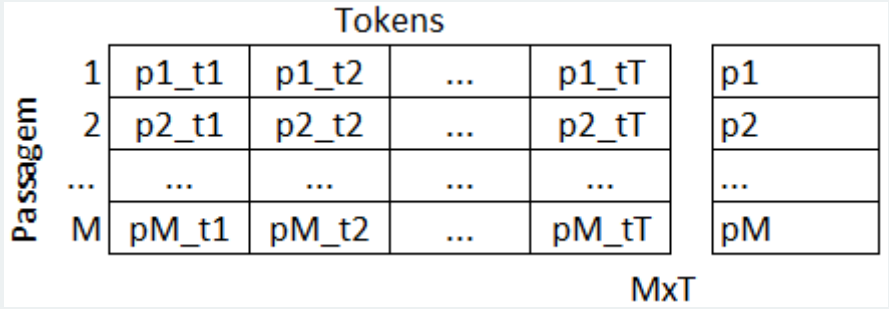
Há dois encoders. O primeiro responsável por gerar os embeddings das passagens e, o outro, das queries. Os embeddings são dados pelo T[CLS] (primeiro elemento da última camada).

Contribuições do artigo

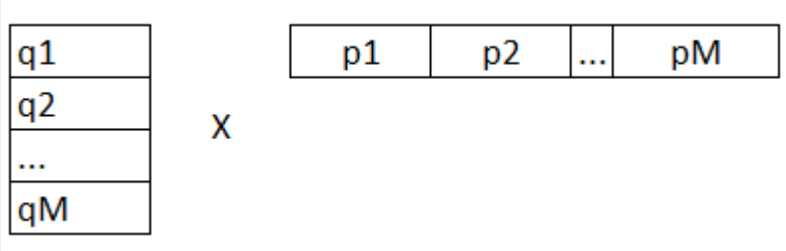
Treinamento:

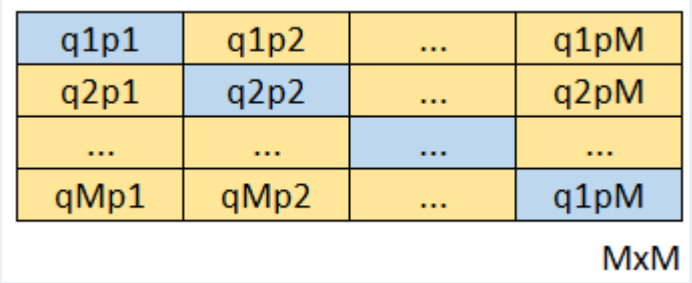
A organização dos dados fica assim:





E a similaridade é calculada como QP^T (no artigo está Q^TP, mas organizei diferente):





Contribuições do artigo

Treinamento:

Cálculo da loss:

$$L(q_i, p_i^+, p_{i,1}^-, \cdots, p_{i,n}^-) \quad (2)$$

$$= -\log \frac{e^{\mathrm{sim}(q_i, p_i^+)}}{e^{\mathrm{sim}(q_i, p_i^+)} + \sum_{j=1}^{n} e^{\mathrm{sim}(q_i, p_{i,j}^-)}}.$$

q1p1	q1p2	...	q1pM
q2p1	q2p2	...	q2pM
...	...	...	...
qMp1	qMp2	...	q1pM

MxM

Usa essa loss pra fazer o fine-tuning de dois encoders.

Contribuições do artigo

Inferência:

Etapa offline - Gera os embeddings de todos os documentos e guarda.

Quando uma query for apresentada, usa o encoder de queries para calcular o vetor de query.

O score é calculado como QP^T , da mesma forma que o cálculo da similaridade na loss.

Obrigado

Leandro Carísio

carisio@gmail.com