NEURAL INFORMATION RETRIEVAL

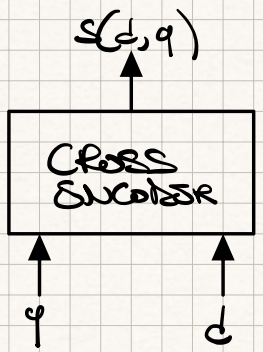
Il task principale in IR è l'ad-hoc relevance ranking.
Per farlo fare a un modello bisogna distinguere:

1) INPUT: token della query e collezione di docs.
2) OUTPUT: Score $S(q,d) \in \mathbb{R}$

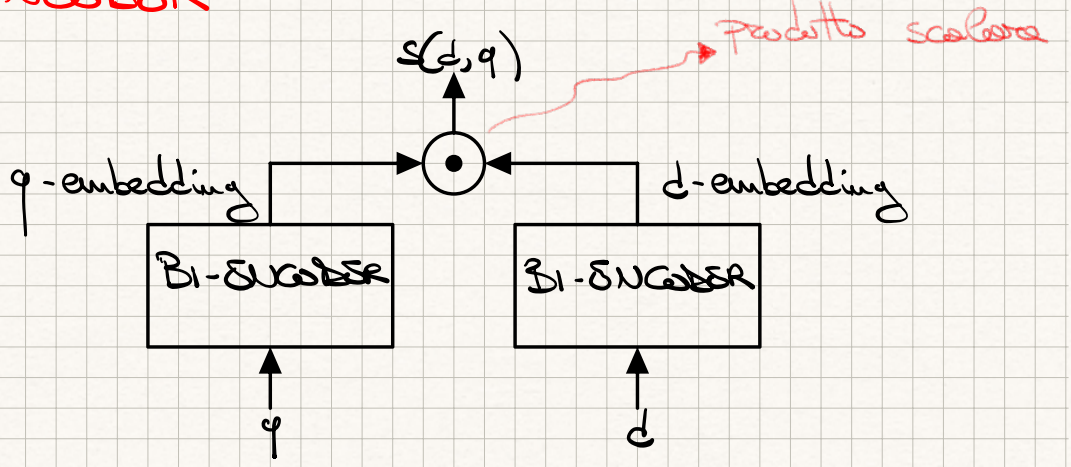
Il modello principe, usato per questo scopo è BERT.

APPROCCI

1) CROSS-ENCODER



2) BI-ENCODER

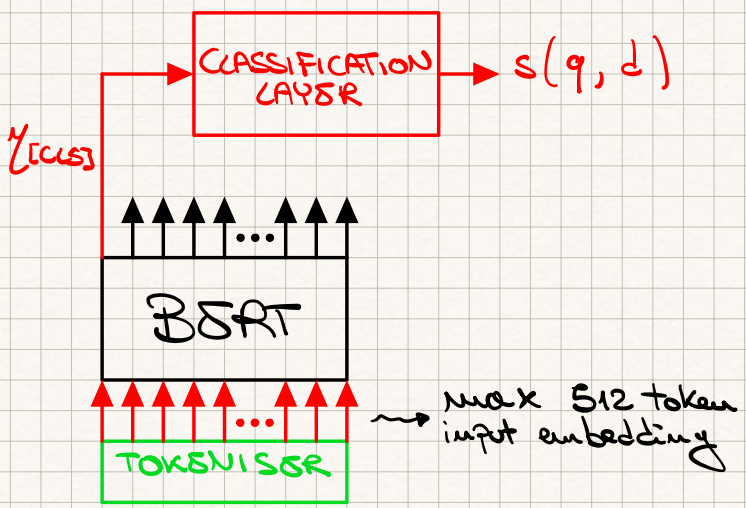


Gli embeddings non sono normalizzati a priori.

Supponiamo di avere a disposizione il miglior modello, come lo usiamo? Ogni volta che arriva una query, devo dare in input al modello tutte le coppie $<q, d_i>$ e sfortunatamente valutare, ci metteremo una vita a raggiungere un risultato...
Difatti non si danno al modello tutte le coppie $<q, d_i>$ ma un loro sotto-insieme, ovvero solo i top-K documenti ricavati usando una score function (TFIDF, BM25, ...) e l'inverted index. Questo approccio si usa per il cross-encoder.

Per il bi-encoder, dato che non vuole le coppie $<q, d_i>$ ma, divide l'ingresso, basta pre-calcolarsi i risultati dei documenti, e sommarli quando arriva una nuova query.

CROSS-ENCODER



Il classification layer in letteratura viene implementato in due modi diversi:

1) Classificatore degenere – NON TROPPO USATO

2) $\gamma_{[CLS]}$ → 768 dim → W → 2 dim $[z_0, z_1]$ → Softmax → 2 dim $[p_0, p_1]$ → $s(q,d) = p_1$

EQUAZIONI CROSS-ENCODER

$$\gamma_{CLS} = BERT(q, d)$$

$$[z_0, z_1] = \gamma_{CLS} \cdot W$$

$$[p_0, p_1] = softmax([z_0, z_1])$$

$$s(q, d) = p_1$$

Basta usare solo il CLS perché, dato che mischiamo q e d insieme, l'embeddings del CLS contiene il "contesto" sia di q che di d.

FINE-TUNING CROSS-ENCODER

Usiamo delle training istances così fatte:

$$\forall q \quad (q, d^+, d^-) \in D$$

Loss

$$\mathcal{L}^{CE} = \delta\left[\ell(q, d^+, d^-) \right]$$

Usiamo la binary cross-entropy:

$$\ell(y, (q,d)) = \begin{cases} -\log(s(q,d|\theta)) & \text{per } y = + \\ -\log(1 - s(q,d|\theta)) & \text{per } y = - \end{cases}$$

Problema: Se l'input è più grande dei 512 token supportati da BERT? Bisogna dividere il documento in parti, dette passaggi, dove uno score a ogni passaggio poi decidere come usare questo pool di score per documento (max score, average score, first score ...)

Come divido il testo del documento? Usiamo la tecnica del Passaging, dove prendiamo parti di documento in modo da sovrapporsi:

Facciamo lo score di ogni passaggio $P_1 \dots P_M$:

$$\left.\begin{array}{l} s(q, P_1) \\ \vdots \\ s(q, P_M) \end{array}\right\} \text{pooling} \rightarrow s(q, d)$$

Pooling:

1) Max Score
2) Avg Score
3) First score

BI-ENCODER

È l'approccio più utilizzato, dato che separa computazionalmente query e documenti.

EQUAZIONI

$$\phi(q) = [\phi_{CLS}, \phi_1, \ldots, \phi_{|q|}] = BERT(q)$$
$$\psi(d) = [\psi_{CLS}, \psi_1, \ldots, \psi_{|d|}] = BERT(d)$$
$$s(q,d) = f(\phi(q), \psi(d))$$

Se la funzione $f(\cdot)$ usa solo ϕ_{CLS} e ψ_{CLS} si chiama **single-representation bi-encoder**. I più famosi sono

1) DPR (2020)
2) ANCE (2021)
3) STAR (2021)

} Usano il prodotto scalare $q \times d$.

Se $f(\cdot)$ usa anche le altre componenti abbiamo un **multi-representation bi-encoder**. Il più famoso è:

1) ColBERT (2020)

ColBERT non usa il prodotto scalare, ma un meccanismo detto **late-interaction** (o **max sim operator**):

$$s(q,d) = \sum_{i=0}^{|q|} \max_{j=0\ldots|d|} (\phi_i \cdot \psi_j)$$