



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií



Vyhledávání

a deep learning

František Kynych
4. 11. 2021 | MVD





Opakování z minulé přednášky

TF-IDF

- Výsledná rovnice při použití TF-IDF a počítání relevance pomocí skalárního součinu:

$$\text{sim}(q, d) = \sum_{i=1}^N x_i y_i = \sum_{w \in q \cap d} c(w, q) c(w, d) \log\left(\frac{M + 1}{df(w)}\right)$$

M = celkový počet dokumentů v kolekci

$df(w)$ = celkový počet dokumentů obsahující slovo w (document frequency)

Okapi BM25

- Výsledná rovnice BM25 při počítání relevance pomocí skalárního součinu:

$$\text{sim}(q, d) = \sum_{i=1}^N x_i y_i = \sum_{w \in q \cap d} c(w, q) \frac{(k+1)c(w, d)}{c(w, d) + k(1 - b + b \frac{|d|}{\text{avdl}})} \log\left(\frac{M+1}{df(w)}\right)$$

- V praxi se používá rozšířený BM25
 - BM25F pro strukturované dokumenty (F = fields)
 - BM25+ zabraňuje příliš velké penalizaci dlouhých dokumentů
- BM25 používá defaultně [Elasticsearch](#)

Limitace předchozích metod

- Bag-of-words
 - Rozdělení textu na jednotlivá slova
 - Nezáleží na pořadí slov v dotazu
- Fungují hůře při vyhledávání nad delšími dokumenty
- Problém při špatném definování dotazu
 - Nelze nalézt synonyma
- Pokud není potřeba řešit některý ze zmíněných problémů, tak je BM25(F/+) dostačující
 - Jinak se používá jako baseline



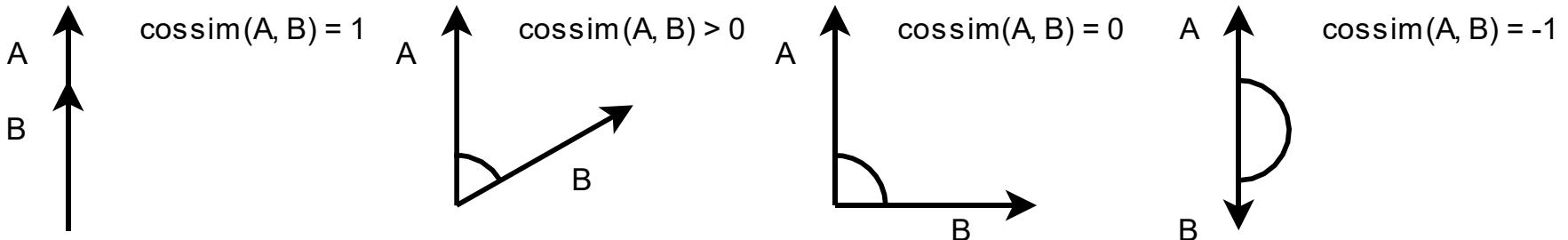
Část I.: Vylepšení vyhledávání a aplikace neuronových sítí

Kosinová podobnost

- Nezávislá na velikosti dokumentů

$$\text{cossim}(A, B) = \cos \phi = \frac{A \cdot B}{\|A\| \|B\|}$$

$$\|A\| = \sqrt{\sum_{i=1}^N A_i^2}$$



Základní vylepšení vyhledávání

1. Použití Word2Vec modelu místo TF-IDF/BM25
2. Generování synonym
3. Generování alternativních dotazů
4. Našeptávání uživateli

Word2Vec retrieval model

- Porovnání vektoru dotazu a dokumentu
- Jak tyto vektory vytvořit?
 - Průměr vektorů slov
 - Stejný problém jako na počátku – jedno četné slovo může ovlivnit výsledek
 - Funguje dobře při velkém množství trénovacích dat
- Word2Vec v kombinaci s TF-IDF vyhlazováním
 - Funguje lépe při menším množství dat
 - Průměrujeme vážené Word2Vec vektory
 - Váhou každého slova je získané TF-IDF, např.:

$$query_{vec} = \frac{1}{|q|} \sum_{w \in q} word2vec(w) * tfidf(w)$$

Generování synonym

- Vytvoření dalších dotazů obsahujících synonyma
 - Např. aeroplane -> plane, airplane, aircraft
 - Větší šance navrácení správného výsledku
- Rozšíření tokenizéru pro dotazy
- Základní přístupy:
 - Vytvoření souboru obsahujících synonyma
 - Složité na tvorbu a údržbu
 - Použití již vytvořeného slovníku synonym pro daný jazyk
 - Např. [WordNet](#) pro angličtinu
 - Neexistuje pro každý jazyk
- Problém těchto přístupů:
 - Chybí kontext
 - Slang
 - Zkratky

Generování synonym

- Chceme nalézt nejbližšího souseda ke slovu analýzou jeho kontextu
 - Nemusí se jednat o synonymum z gramatického hlediska
 - Založeno na distribuční hypotéze
- Použití vektorové reprezentace slov – Word2Vec
 - Preferován skip-gram přístup
 - Lepší pro málo četná slova
 - Hledáme slova s největší kosinovou podobností

Generování synonym – Word2Vec

- U generování synonym je potřeba rozmyslet přístup k dané aplikaci
- Možnosti:
 - Generování synonym při indexování každého dokumentu
 - Při dlouhých dokumentech zabere velké množství místa
 - Generování synonym jen pro některé slovní druhy
 - Potřeba implementace part of speech (PoS) taggingu (podstatná slova, slovesa)
 - Zaměření na krátké dokumenty
 - Mají menší pravděpodobnost shody s dotazem
 - Zaměření pouze na častá slova
 - Generování synonym s vyšším prahem (pouze nejbližší slova)

Generování synonym – Word2Vec

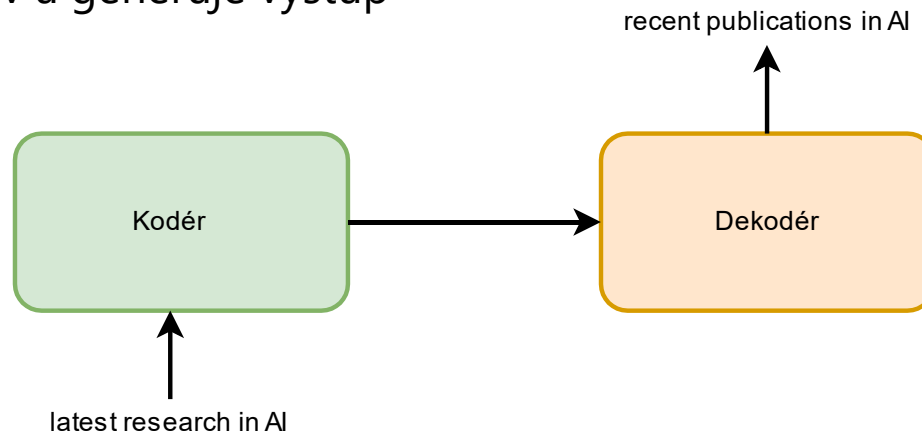
- V produkci je potřeba pravidelně aktualizovat
- Pozor na antonyma (slova opačného významu)
 - Při menším množství trénovacích dat mohou mít antonyma velkou podobnost
 - Je potřeba otestovat pro cílovou aplikaci

Generování alternativních dotazů

- Označováno jako *automatic query expansion*
- Cílem je vygenerovat alternativní dotaz, který je sémanticky stejný
 - Chceme minimalizovat výskyt dotazů bez výsledků
- Např.: books about artificial intelligence -> publications from the field of artificial intelligence
- Kde získat data?
 - Procházení logů vyhledávání od uživatelů
 - Dotazy, které vrátí podobné výsledky
 - Podobné dotazy od uživatele v daném časovém intervalu
 - Přidání textu z nalezených relevantních dokumentů

Generování alternativních dotazů

- Využití rekurentních neuronových sítí
 - Vysvětleno využitím Seq2Seq architektury
- Seq2Seq architektura se skládá z dvou částí
 - **Kodér**
 - Zpracuje vstupní informaci a poté předá skrytý stav do dekodéru
 - **Dekodér**
 - Přijme skrytý stav a generuje výstup



Generování alternativních dotazů

- Vstupní sekvence se ukončuje end-of-sequence tokenem (<EOS>)
 - Poté se začne generovat výstup
- Skrytý stav se zde také označuje jako myšlenkový vektor uživatele
- Dnes je lepší používat [Transformer](#) nebo [novější architektury](#)

Našeptávání uživateli

- Chceme, aby uživatel psal lepší dotazy
 - Autocomplete
 - Méně dotazů s žádnými výsledky nebo s malou relevancí výsledků
- Kde získat data?
 - Slovníky důležitých slov (např. nejčastěji hledané)
 - Dříve použité dotazy
 - Použití indexovaných dokumentů (např. titulky)

Našeptávání uživateli

- Použití slovníků
 - Prohledávání prefixového stromu
- Vyzkoušení vyhledávání na Googlu: „books about search“
 - Autocomplete postupně provádí:
 - Dokončení samostatných slov
 - Doporučení více slov najednou (nebo frází)
 - Odstraňuje některá napsaná slova (stopword filtr)
 - Nahrazuje některá slova
 - Při delším dotazu vynechává prefixová slova
- Často defaultně implementováno pomocí konečných automatů
 - Např. v třídě AnalyzingSugester v Apache Lucene

Našeptávání uživateli

- N-gram jazykové modely
 - Vychází z Markovovy vlastnosti
 - Pravděpodobnost budoucího slova závisí na přechozích slovech
- Použití bigramu (n=2)
 - Pravděpodobnost dalšího slova závisí na současném slovu

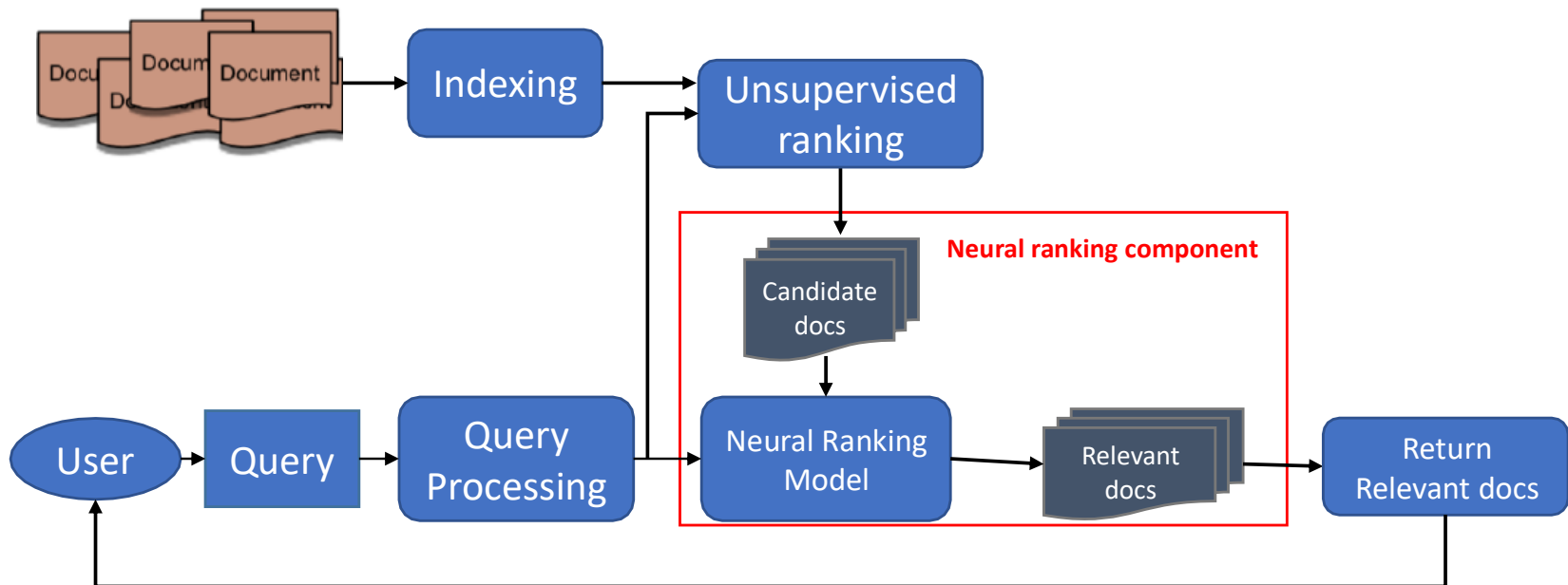
$$P(\textit{books about search}) = P(\textit{about}|\textit{books}) \times P(\textit{search}|\textit{about})$$

- Použito v třídě FreeTextSugester v Apache Lucene

Našeptávání uživateli

- Využití rekurentních neuronových sítí
 - Chceme, aby síť doplňovala uživatelský dotaz
- Rozšíření Word2Vec modelem
 - Natrénován na stejných datech jako našeptávání
 - Vytvoří alternativní dotaz na základě nejmenší vzdálenosti

Standardní přístup k vyhledávání



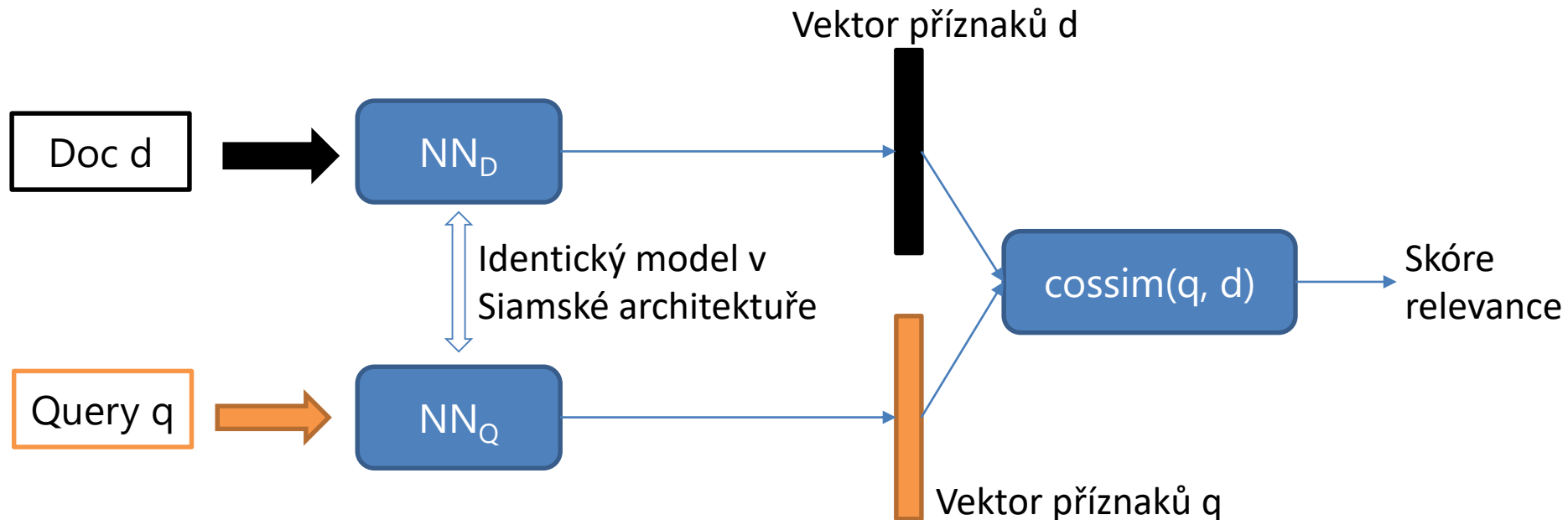
<https://arxiv.org/pdf/2102.11903.pdf>

Způsoby učení neuronové sítě

- Pointwise
 - Dataset dotazů a dokumentů s cílovým hodnocením pro každý dokument
 - Neuronová síť se učí správně ohodnotit dokumenty
 - Složitější vytvoření datasetu
- Pairwise
 - Funguje na principu binární klasifikace
 - Učí se určit, který ze dvou dokumentů je relevantnější
 - Více citlivá metoda na šum
 - Při špatném ohodnocení jednoho dokumentu může selhat i hodnocení všech dalších
- Ad-hoc (Listwise)
 - Dataset dotazů s listem cílových dokumentů
 - Neuronová síť se učí optimalizovat některou z metrik pro vyhledávání (např. NDCG) nebo se snaží modelovat pravděpodobnostní rozložení permutace

Representation-focused models

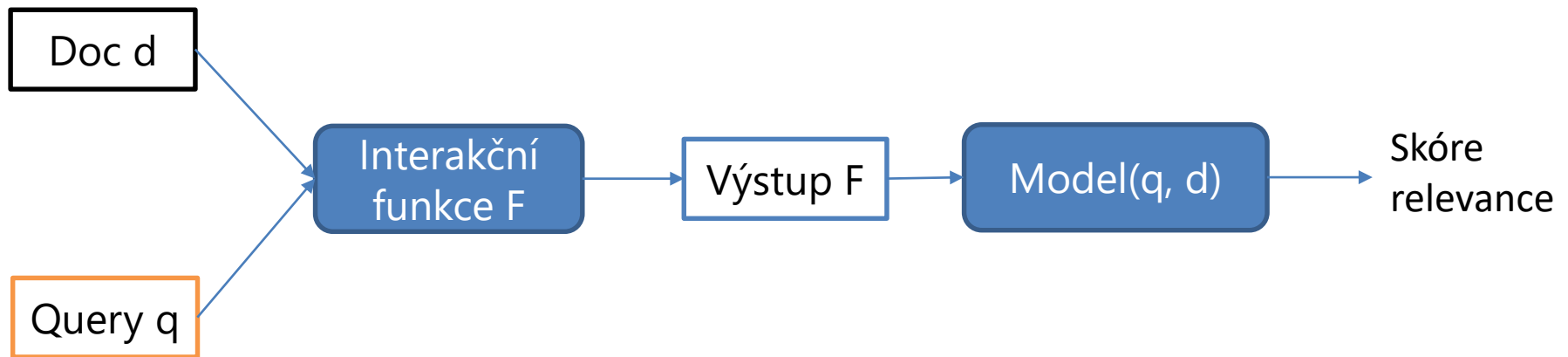
- Modely extrahují informaci z dokumentu a z dotazu
 - Poté se použije kosinova podobnost pro hodnocení relevance
- Většinou tzv. siamská architektura



- Použití např. [Convolution deep structured semantic modelu](#)

Interaction-focused modely

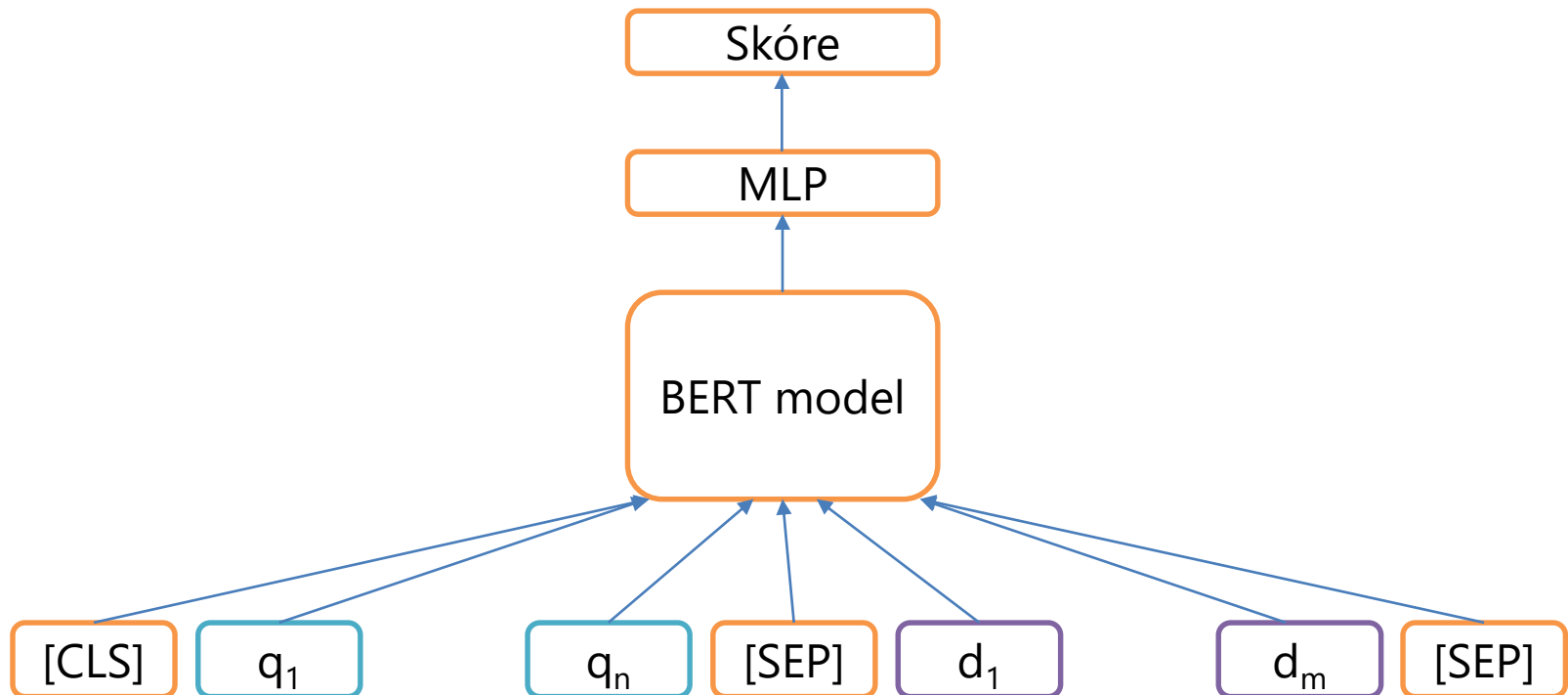
- Modely, do kterých vstupuje najednou dotaz i dokument
- Interakční funkce
 - Model, který se snaží zachycovat vztah mezi dokumentem a dotazem (např. společná slova)
- Výstupní model počítá relevanci na základě příznaků z interakční funkce
- Např. [Deep Relevance Matching Model](#)



Využití předtrénovaných jazykových modelů

- Využití ELMO / BERT předtrénovaného modelu
 - Fine-tuning pro konečnou aplikaci
- BERT
 - Maximálně 512 vstupních tokenů
 - Dokument lze rozdělit na komponenty (věty)
 - Relevance dokumentu odpovídá top-k nejrelevantnějším komponentám
 - => Výpočet relevance přes páry (dotaz, komponenta)
 - => Relevance je rovna vážené sumě top-k výsledků
 - Alternativní přístupy
 - Získání query a dokument embeddingu => sjednocení a výpočet skóre
 - Fúze BERT modelu a jiného modelu pro výpočet relevance
- TransformerXL pro dlouhé dokumenty

Využití předtrénovaných jazykových modelů



Vícejazyčné vyhledávání

- Rozšíření standardního vyhledávání
 - Detekce jazyka
 - LSTM neuronová síť
 - Strojový překlad
 - Transformer modely
 - Vícejazyčný embedding
 - Transformer modely (2021, Facebook AI)
- Překlad dotazu vs dokumentu

Užitečná literatura / kurzy

- TEOFILI, Tommaso. *Deep learning for search*. Shelter Island: Manning, [2019]. ISBN 978-161-7294-792.
- [Neural Ranking Models For Document Retrieval \(2021\)](#)
 - 2021 – aktuální shrnutí modelů neuronových sítí používaných pro hodnocení dokumentů