Pristopi k iskanju informacij - od klasičnih do jezikovnih modelov

Andrej Erjavec

May 2025

Ključne besede: Iskanje informacij, vgnezditve, Word2Vec

1 Uvod

Z naraščajočo količino digitalno shranjenih podatkov postaja učinkovito iskanje informacij eden ključnih izzivov sodobne informacijske tehnologije. Vse od začetka digitalnega shranjevanja podatkov obstaja potreba po iskanju želenih informacij znotraj teh, zaradi česar se je v domeni računalniške znanosti razvila veja iskanja podatkov (angl. Information Retrieval - IR).

Velik izziv na tem področju predstavljajo nestrukturirani podatki, kot so slike, video in besedilo - količina teh v zadnjem desetletju predvsem zaradi množične uporabe družbenih omrežij ter senzorskih podatkov IoT naprav že močno presega količino strukturiranih podatkov. Več kot 80 % vseh podatkov, shranjenih v informacijskih sistemih, je nestrukturiranih [10]. S Pojmom *nestrukturirani podatki* opisujemo podatke, ki nimajo vnaprej definiranega modela ali sheme, zato njihov zapis v klasično relacijsko podatkovno bazo ni mogoč [6]. V primeru relacijskih podatkovnih baz je iskanje podatkov ter razumevanje relacij med njimi enostavno, kar pa ne velja za nestrukturirane podatke, katerih procesiranje posledično zahteva namenske metode ter algoritme. V domeni iskanja informacij se je zato skozi zgodovino razvilo več različnih pristopov k reševanju tega problema - od tradicionalnih statističnih modelov kot je TF-IDF so se s povečevanjem procesnih ter shrambnih zmogljivosti računalniške opreme, razvijale vse bolj napredne metode za procesiranje podatkov. V zadnjem desetletju so se z razvojem umetne inteligence ter strojnega učenja, možnosti uporabe teh tehnologij rezširile tudi na področje procesiranja besedil, kjer omogočajo hitrejše ter bolj učinkovito iskanje informacij [8].

V tem članku bomo obravnavali ključne izzive, ki predstavljajo temeljno motivacijo za raziskave na področju iskanja informacij, ter predstavili glavne modele in njihove implementacije, namenjene učinkovitemu procesiranju podatkov. Pri tem se bomo osredotočili na precesiranje nestrukturiranih besedilnih podatkov.

2 Iskanje informacij - Information retrieval

Iskanje informacij (angl. Information retrieval) je veja računalniške znanosti, ki se ukvarja s problemom iskanja relavantnih informacij iz obsežnih zbirk dokumentov. IR je proces, ki vključuje hranjenje, reprezentacijo in zmožnost iskanja informacij v množici nestrukturiranih ali pol-strukturiranih dokumentov za namen odkrivanja znanja [5, 13]. Pri iskanju informacij gre najpogosteje za besedilne dokumente, zato lahko v tem primeru govorimo o dokumentnih sistemih (angl. Document Management Systems). Zaradi potreb po obdelavi besedila, se ta veja tesno povezuje s področjema naravne obdelave jezika (angl. Natural Language Processing - NLP) ter rudarjenja besedil (angl. Text Mining). Glavna naloga procesa iskanja informacij je na podlagi uporabnikove poizvedbe (angl. query) poiskati podmnožico najbolj relavantnih objektov, ki ustrezajo poizvedbi. Ker poizvedbi pogosto ustreza več objektov, so ti rangirani glede na stopnjo ujemanja s poizvedbo, pri čemer so uporabljene različne funkcije za izračun podobnosti. Sistem vrne podmnožico z najboljšim ujemanjem.

V splošnem je proces iskanja informacij sestavljen iz več faz. V nadaljevanju bomo predstavili faze, ki so vezne na proceisranje nestrukturiranih besedil. Za boljšo predstavitev strukture procesa, bomo v tem članku proces razdelili na dve fazi: priprava dokumentov na iskanje ter iskanje. V prvo fazo uporabnik sistema ni neposredno vključen, med tem ko pri drugi fazi sistemu poda poizvedbo, ta pa mu vrne odgovor kot podmnožico relavantnih dokumentov.

2.1 Priprava na iskanje

Predobdelava (angl. preprocessing) Ko sistem zbere množico dokumentov, se izvede predobdelava, ki vključuje več korakov in metod [15]. Za poenotenje besedila je najenostavnješi korak sprememba črk iz velikih v male. S tem se izognemo, da bi računalnik tretiral pomensko enake besede z veliko ali malo začetnico kot pomensko različne. V procesu tokenizacije se vhodno besedilo razbije na logične enote, imenovane *tokene*, ki so lahko odvisno od potreb sistema določeni kot posamezne besede, stavki ali druge logične enote besedila. V naslednjem koraku se odstranijo pomensko manj pomembne besede kot so “the”,” are”, ”is”, ”and”, saj lahko negativno vplivajo na kasnejšo klasifikacijo.

Z metodo *stemming* se besedam odstrani končnica tako, da dobljena osnovna beseda vseeno ohrani svoj semantični pomen. Primer: Studying → Study. Rezultat *stemming* metode ni vedno pomenska beseda.

Podobna metoda je *lematizacija*, ki odstrani ali zamenja končnico besede tako, da je rezultat vedno pomenska beseda. Primer: Caring → Care.

Indeksiranje Struktura vhodnih dokumentov običajno ne omogoča učiknovitega iskanja informacij, zato jih je potrebno v procesu indeksiranja pretvoriti v obliko, ki to omogoča in je razumljiva računalniku. Indeksiranje se izvaja na nivoju tokenov, generiranih v predhodnem procesu tokenizacije, glavni namen pa je

pretvorba besedilne vsebine v podatkovno strukturo, ki omogoča hitro lociranje dokumentov, ki vsebujejo izraze poizvedbe. Najpreprostejša oblika indeksa je inverzni indeks [1] - struktura podobna razprešeni tabeli (angl. hash table), ki za vsak token hrani nabor dokumentov, ki ta token vsebujejo, oziroma lokacijo tokena znotraj dokumenta. Kot alternativa klasičnim metodam indeksiranja so se kasneje razvile vgnezditve (angl. embeddings), ki omogočajo predstavitev in primerjavo dokumentov v večdimenzionalnem vektorskem prostoru.

2.2 Iskanje

Iskanje je glavna faza v procesu IR in prva faza, v katero je vključen uporabnik sistema. Ta faza se začne z uporabnikovov potrebo po informacijah, ki je nato formulirana v obliki poizvedbe v naravnem jeziku. Po vnosu poizvedbe sistem najde nabor dokumentov, ki poizvedbi najbolj ustrezajo glede na vsebino besedila. Več različnih pristopov in algoritmov omogoča primerjavo poizvedbe z vsebino dokumentov. Najosnovnejši pristopi temeljijo na neposrednem primerjanju poizvedbe z besedilom dokumenta, pri čemer sistem išče popolna ujemanja terminov. Tovrstna ad-hoc primerjava je sicer trivialna za implementacijo, vendar se pogosto izkaže za neučinkovito. Zaradi razlik v izražanju avtorjev besedila ter uporabnikov, popolna ujemanja redko zadostujejo za kakovostno iskanje. Ad-hoc primerjavo je sicer mogoče izboljšati z uporabo regularnih izrazov, vendar je izdelava regularnih izrazov, ki pokrijejo vse možnosti uporabnikovih poizvedb izjemno zahtevna.

Uspešnost iskanja se izboljša pri uporabi algoritmov za izračun leksikalne podobnosti - Levenshteinova razdalja ali Hammingova razdalja sta primer algoritmov, ki omogočata primerjavo nizov na osnovi minimalnega števila znakovnih operacij (vstavljanj, brisanj ali zamenjav), potrebnih za pretvorbo enega izraza v drugega.

Omenjene preproste metode iskanja so se izkazale kot neučinkovite, zaradi česar so se raziskave usmerile na iskanje s pomočjo tehnik clustering-a, kjer so logične enote besedila pretvorjene v vektorje, podobnost med njimi pa je izračunana kot vektorska razdalja (kosinusna ali evklidska razdalja). Od tradicionalnih vektorskih modelov, od katerih je glavni predstavnik TF-IDF, so v zadnjem desetletju v ospredje prišli modeli, ki temeljijo na nevronskih mrežah in vgnezditvah (angl. embeddings). Med njimi je vredno omeniti Word2Vec ter ostale modele kot sta GloVe in BERT.

Rangiranje rezultatov Ko je nabor kandidatnih dokumentov identificiran, sledi rangiranje. V prejšnji fazi sistem izbere podmnožico dokumentov, ki ustrezajo poizvedbi, rangirni modeli pa to podmnožico uredijo glede na relavantnost ter vrnejo N najustreznejših.

2.3 Izzivi pri iskanju informacij

V fazi iskanja uporabnik sistemu poda poizvedbo, preko katere predstavi svoje omejeno področje interesa v povezavi z vsebino dokumentov. Z vidika uporabniše izkušnje se je kot dobra praksa uveljavila možnost vnosa poizvedbe v naravnem jeziku preko iskalnega polja [14]. Kljub temu pa tak pristop predstavlja tudi enega ključnih izzivov pri zasnovi sistemov za iskanje informacij iz nestrukturiranih podatkov.

Uporabniki namreč običajno ne poznajo natančne strukture ter terminologije dokumentov, zaradi česar prihaja do semantičnih neskladij med poizvedbami ter vsebino dokumentov. V tem kontekstu se uporaba ad-hoc pristopov za iskanje izkaže kot neučinkovita. Sistemi za iskanje informacij so podvrženi težavam pri uporabi naravnega jezika - polisemantičnost označuje pojav, ko ima lahko ena beseda več različnih pomenov. Enostavno ujemanje ključnih besed brez upoštevanja pomena lahko zato vodi do napačne interpretacije poizvedbe. Težavo je mogoče omejiti s podajanjem kontekstualnih informacij, s čimer se omeji pomen polisemantičnih besed na eno samo področje. Poleg tega velik izziv predstavlja tudi problem neskladja v besedišču (angl. Vocabulary mismatch problem) [16] oziroma pojav, ko različni uporabniki za isti koncept uporabljajo različna poimenovanja.

V primeru nestrukturiranih besedil so podatki pogosto razpršeni po besedilu ter povezani zgolj s semantiko besedila. Definiranje relacij med koncepti preko semantike močno oteži iskanje ter povezovanje relavantnih informacij. Zaradi uporabnikovega nepoznavanja strukture dokumentov, ne obstaja enoličen način za kreiranje poizvedb, kakor to velja v relacijskih podatkovnih bazah.

Učinkovit sistem za iskanje informacij mora biti prilagojen navedenim izzivom tako, da zmanjšuje negativne vplive lastnosti naravnega jezika na natančnost rezultatov poizvedovanja.

V nadaljevanju najprej predstavimo teoretične modele za iskanje informacij, njihove implementacije v praktične algoritme ter področja uporabe.

3 Modeli za iskanje informacij

Z naraščajočimi potrebami po hitrem in učinkovitem iskanju informacij v rastočem številu nestrukturiranih podatkov, se je skozi zgodovino uveljavilo več modelov, ki predlagajo načine za predstavitev ter primerjavo podatkov. V tem poglavju začnemo s klasičnimi Boolovimi modeli, nadaljujemo s probabilističnimi modeli ter modeli v vektorskem prostoru. Nazadnje opišemo novejše pristope, ki temeljijo na nevronskih mrežah in velikih jezikovnih modelih.

3.1 Boolov model in razširjeni Boolov model

Boolov model, predstavljen v 50ih letih prejšnjega stoletja, je najstarejši izmed omenjenih in temelji na neposredni primerjavi med poizvedbo in vsebino dokumentov. Ta model definira indeksiranje dokumentov s ključnimi besedami glede na njihovo vsebino, pri čemer iskanje poteka z nizanjem izrazov, med seboj

povezanih z logičnimi operatorji AND, OR in NOT. Kombiniranje ključnih besed z logičnimi operatorji predstavlja poizvedbo, ki definira nabor vsebinsko ustreznih dokumentov. Poizvedba *"information theory"* AND *"neural netowrks"* na primer vrne dokumente, ki so indeksirani z obema omenjenima izrazoma. Ker gre za logične operacije, lahko ciljni nabor dokumentov predstavimo z Venn diagramom, kjer vsak obroč diagrama predstavlja en izraz oziroma ključno besedo.

Prednost takšnega modela je njegova preprosta implementacija ter enostavnost uporabe za izkušene uporabnike, ki razumejo delovanje Boolove logike in so sposobni popraviti formulacijo poizvedbe v primeru nepričakovanega rezultata [4]. Boolova logika omogoča visoko stopnjo nadzora nad množico pridobljenih dokumentov, poleg tega pa zagotavlja jasno in lahko berljivo strukturo poizvedb. Klasični Boolov model je deterministične narave - tako kot je rezultat logičnega izraza je tudi rezultat poizvedbe binaren - dokument ustreza poizvedbi ali pa ji ne ustreza. Pri tem je glavna težava odsotnost mehanizma za rangiranje dokumentov glede na ustreznost. Ker izrazovv poizvedbi ni mogoče določiti uteži pomembnosti, lahko rezultat odstopa od pričakovanega v primeru, ko dokument striktno ne vsebuje vseh v poizvedbi podanih izrazov. V izogib tej pomanjlkjivosti je bila razvita izboljšana različica v obliki razširjenega Boolovega modela.

Razširjeni Boolov model ohranja strukturo klasičnega modela z dodatkom algoritma za rangiranje dokumentov. Kljub več različnim pristopom k rangiranju, se je za najbolj uspešnega izkazal P-norm model [12]. Namesto stroge binarne klasifikacije (relevantno/nerelavantno), model omogoča rangiranje dokumentov glede na stopnjo njihove podobnosti s poizvedbo. Za ta namen se uporablja p-norm metrika, ki omogoča interpolacijo med strogo logiko in vektorjem podobnim pristopom. Za poizvedbo oblike "A AND B" se podobnost med dokumentom in poizvedbo izrazi kot:

$$sim(D, A\ AND_p\ B) = 1 - \left(\frac{(1 - d_A)^p + (1 - d_B)^p}{2} \right)^{1/p}$$

kjer D_A in d_B predstavljata uteži (npr. TF-IDF) za termina A in B v dokumentu. Manjše vrednosti p omogočajo večjo toleranco do delne ujemljivosti, medtem ko $p \to \infty$ model konvergira k klasični Boolovi logiki.

3.2 Probabilistični model

Podobno kot razširjeni Boolov model tudi probabilistični model stremi k rangiranju dokumentov. Tako dokumenti kot poizvedbe so predstavljene kot binarni vektorji. Vsaka komponenta vektorja predstavlja izraz ali lastnost dokumenta in v binarnem zapisu nakazuje ali dokument vsebuje ta izraz ali ne. Model predpostavlja, da je pojavljanje izrazov v besedilu statistično neodvisno. Model uvaja pojem relavantnosti, kjer R označuje relavanten dokument ter $\bar{R}$ nerelavanten dokument. Za vsak dokument želimo izračunati $P(R \mid d, q)$, kar pomeni verjetnost, da je dokument d relavanten za poizvedbo q. Z uporabo Bayesovega

pravila lahko za vsak izraz v poizvedbi izračunamo razmerje, kako bolj verjetno je, da se ta izraz pojavlja v relevantnih dokumentih kot v nerelevantnih. Skozi čas se je razvilo več različnih praktičnih implementacij tega modela, ki so poskušale izboljšati klasični TF-IDF model uteževanja. Izmed teh se je za najbolj uporabnega izkazal `BM25`, ki kombinira *tf*, *idf* ter poleg tega normalizira dolžino besedila, s čimer se prepreči prevlado daljših besedil zgolj zaradi večjega števila besed.

3.3 Vektorski modeli

Vektorski modeli so eden najpomembnejših pristopov pri iskanju informacij in so temelj številnih sodobnih informacijskih iskalnikov. Glavna ideja modela je predstavitev dokumentov ter poizvedb v večdimenzionalnem Evklidskem vektorskem prostoru. Za vse vektorske modele je zančilen enak proces - dokumenti so predstavljeni kot vektorji. Ko uporabnik poda poizvedbo, je tudi ta pretvorjena v vektor ter primerjana z vektorji dokumentov. Vsaka dimenzija vektorja predstavlja eno besedo iz korpusa, torej je dimenzija enaka številu vseh različnih besed v besedilu. V tem primeru govorimo o redki predstavitvi vektorjev (angl. sparse vector representation) oziroma one-hot kodiranju, kjer je večina vrednosti v vektorju ničelnih. To je posebej značilno za primere, ko dokument vsebuje le majhno podmnožico besed iz slovarja. V primeru, ko komponenta vektorja predstavlja prisotnost oziroma odsotnost izraza, je vrednost te komponente binarna. Lahko pa je predstavljena tudi kot vrednost TF-IDF ali bolj enostavno kot frekvenca pojavitve izraza v dokumentu. Vektorski modeli omogočajo primerjavo z delnim ujemanjem - dokumenti, ki vsebujejo več skupnih izrazov, bodo v vektorskem prostoru bližne skupaj oziroma bo razdalja med njimi manjša. Za primerjavo redkih vektorjev je najpogosteje uporabljena mera za izračun razdalje med njimi kosinusna podobnost (angl. cosine similarity).

$$sim(\vec{d}, \vec{q}) = \frac{\vec{d} \cdot \vec{q}}{\|\vec{d}\| \cdot \|\vec{q}\|} \tag{1}$$

V nadaljevanju kronološko predstavimo nekaj najbolj poznanih modelov za pretvorbo besedila v vektorje ter opišemo osnove njihovega delovanja.

Bag of Words je prvi in najbolj enostaven model za vektorsko predstavitev besedil. Glavna ideja modela je, da dokument obravnava kot neurejen skupek besed, pri čemer ignorira slovnični vrstni red, skladnjo in semantiko. Vsak dokument je predstavljen kot vektor besedne frekvence, kjer vsaka komponenta ustreza določeni besedi iz slovarja. Vrednost komponente vektorja je pogosto bodisi binarna (prisotnost/odsotnost besede) bodisi število pojavitev besede v dokumentu.

TF-IDF (Term Frequency — Inverse Document Frequency) je nekoliko izboljšan model na osnovi Bag of Words. Vrednost TF-IDF je izračunana kot produkt dveh komponent *tf* in *idf*, kjer *tf* predstavlja število pojavitev neke besede v dokumentu, *idf* pa znižuje težo izrazom, ki se pogosteje pojavljajo v

besedilu (npr. *a*, *that*, *the*) oziroma pove koliko informacije drži vsaka beseda. Formalno zapišemo

$$TF(t,d) = \frac{f_{t,d}}{\sum_k f_{k,d}} \tag{2}$$

kjer $f_{t,d}$ predstavlja število pojavitev izrata t v dokumentu d ter

$$IDF(t) = \log\left(\frac{N}{n_t}\right) \tag{3}$$

kjer je N skupno število dokumentov, n_t pa število dokumentov, ki vsebujejo vsaj eno pojavitev izraza t. Končno vrednost izračunamo kot

$$TFIDF(t,d) = TF(t,d) \cdot IDF(t) \tag{4}$$

Čeprav je TF-IDF učinkovit in enostaven za implementacijo, ima nekatere pomembne omejitve. Ena izmed njih je linearna rast uteži s pogostostjo izraza, kar lahko vodi do prioritizacije izrazov, ki se pogosto pojavljajo znotraj posameznega dokumenta. Prav tako TF-IDF ne vključuje normalizacije dolžine dokumenta, kar lahko povzroči pristranskost v korist daljših besedil.

Za izboljšanje teh pomanjkljivosti je bil razvit model BM25 (Best Matching 25) [11], ki temelji na probabilističnem modelu iskanja informacij. BM25 uvaja izboljšano formulacijo za uteževanje izrazov, ki vključuje logaritemsko funkcijo, normalizacijo dolžine dokumenta in dva hiperparametra za boljšo kontrolo vpliva posameznih komponent.

3.4 Vgnezditve

Do sedaj omenjeni vektorski modeli imajo skupno pomanjkljivost - producirajo vektorje, katerih dimenzija je enaka številu besed v slovarju, zaradi česar so nastali vektorji redki. Poleg tega klasični modeli niso sposobni zajeti informacij o semantiki (pomenu) besed. Besedi *vozilo* in *avto* sta sicer popolnoma različni, a imata zelo podoben pomen. Rešitev za omenjene izzive je bila najdena v predstavitvi besedil z vgnezditvami (angl. embeddings), ki omogočajo predstavitev besed v obliki gostih vektorjev (angl. dense vectors) fiksne dimenzije. Dimenzija vektorjev je sicer odvisna od implementacije modela, a neodvisna od dolžine in pomena besede. Poleg zmanjšane dimenzije vektorjev so vgnezditve uporabne tudi za zajem semnatike besed tako, da so pomensko podobne besede med seboj bližje v vektorskem prostoru.

Word2Vec [7] je eden izmed najbolje poznanih primerov predstavitve z gostimi vektorji. Model je bil razvit s strani Googla leta 2013 in je predstavljal prelomnico na področju semantičnega procesiranja naravnega jezika. Word2Vec omogoča napovedovanje kontekstualnih besed ali pa napovedovanje iskane besede iz konteksta, pri tem pa temelji na pravilu: "Če sta dve besedi semantično podobni, naj bosta tudi njuni vrednosti v vektorksem prostoru podobni". Kontekst besede je zajet preko drsečega kontekstualnega okna, ki zajema okoliške besede okrog ciljne besede. Pretvorba besed v redke vektorje

je implementirana z množenjem goste predstavitve besede z vgnezditveno matriko. Glavni cilj je, da se model nauči te vgnezditvene matrike, za kar Word2Vec uporablja dva pristopa: *CBOW* in *skip-gram*. Oba modela sta v obliki preproste nevronske mreže z vhodnim, vgnezditvenim in izhodnim slojem. CBOW (Continuous Bag of Words) je proces, s katerim poskušamo predvideti ciljno besedo glede na okoliške kontekstualne besede. Pri tem se vse kontekstualne besede pretvorijo v vektorje, nato pa se izračuna njihovo povprečje. Dobljena vrednost je vhod v softmax funkcijo, ki predvidi najbolj verjetne besede, ustrezne glede na kontekst. Pri arhitekturi skip-gram pa gre za obraten proces in sicer za napovedovanje kontekstualnih besed na podlagi ene podane besede.

Word2Vec najbolje deluje na majhnih korpusih besed, pri čemer je omejen na obdelavo besed brez možnosti obdelave večjih besedilnih enot. Prav tako je omejeno kontekstualno okno okoli besede, zato model ni sposoben zajeti konteksta celotnega korpusa.

GloVe (Global Vectors) [9] je model nenadzorovanega učenja, ki izboljša model Word2Vec z uporabo analize so-pojavitve besed za učenje vektorske predstavitve. Pri tem je ključna ideja, da so razmerja med besedami zakodirana v matriki so-pojavitev, ki šteje, kolikokrat se neka beseda pojavi v kontekstu druge besede v celotnem korpusu. Tako Word2Vec kot GloVe nista sposobna učinkovitega reševanja problema polisemije besed, saj ne upošteva lokalnega konteksta besede za ugotavljanje njenega pomena. Zaradi tega polisemantičnim besedam vedno priredi identične vektorje. GloVe sicer zajame širši kontekst, vendar ne omogoča prepoznavanja zahtevnejših jezikovnih struktur kot so odvisnosti med besedami, negacije ter pomen pripon in predpon. V primerjavi z CBOW in Skip-gram modeloma, prisotnima v Word2Vec, na enakem korpusu besedil GloVe doseže boljšo hitost učenja ter znatno izboljša natančnost napovedovanja [9].

3.5 Jezikovni modeli

Z razvojem globokega učenja ter z njim povezanih transformerskih modelov so se odprle nove možnosti na področju procesiranja naravnega jezika. Ena glavnih prodobitev so modeli kontekstualnih vgnezditev, ki omogočajo zajem širšega konteksta za razpoznavanje pomena besed ter stavkov. Eden glavnih predstavnikov transformerskih modelov za procesiranje naravnega jezika je Googlov model BERT.

BERT [3] je jezikovni model za procesiranje naravnega jezika, razvit leta 2018, ki je zaradi svoje zmožnosti procesiranja večjih enot besedila (npr. stavkov) postal jedro Googlovega spletnega iskalnika. V napsrotju s prejšnjimi mideli, ki temeljijo na enosmernem branju besedila, BERT uvaja dvosmerno preverjanje besedila s čimer lahko bere kontekst v obeh smeh hkrati. Model je predhodno treniran na angleških besedilih z Wikipedia in sicer za opravljanje dveh nalog - maskirano jezikovno modeliranje (angl. Masekd language modeling - MLM) in napovedovanje naslednjega stavka (angl. Next entence prediction - NSP). Pri MLM model naključno maskira določene besede v vhodnem zaporedju ter uči mrežo, da jih rekonstruira na podlagi širšega konteksta. Napovedovanje nasled-

njega stavka temelji na algoritmu, ki določi, ali si dva stavka logično sledita. Osrednja prednost modela BERT je njegova zmožnost generiranja kontekstualiziranih vektorskih predstavitev besed, kar pomeni, da se pomen besede dinamično prilagaja glede na okoliško besedilo. To omogoča učinkovito obravnavo jezikovnih pojavov, kot so polisemija, sopomenke, ter razumevanje odnosov med posameznimi besedami.

Zaradi odprtokodne narave je BERT postal osnova za številne domenske različice in nadgradnje, med drugimi tudi Docbert [2] za klasifikacijo dokumentov. Izpeljanke modela BERT so prilagojene specifičnim nalogam in jezikovnim domenam, kar dodatno potrjuje njegovo prilagodljivost in široko uporabnost v raziskovalnem in industrijskem okolju.

Generative pre-trained transformers (GPT) V zadnjih letih smo priča hitrem napredku v razvoju asistentov s temeljem na velikih jezikovnih modelih. S temi v ospredje prihajajo modeli GPT (angl. Generative Pre-trained Transformer), ki temeljijo na transformerski arhitekturi. GPT modeli so v nasprotju z modeli kot je BERT, osredotočeni na generacijo tekočega, semantično smiselnega besedila. V fazi iskanja informacij GPT modeli običajno niso uporabljeni za vektorsko predstavitev dokumentov, temveč se lahko uporabljajo v kasnejši fazi za generiranje odgovorov na poizvedbe v naravnem jeziku, pri čemer upoštevajo vsebino pridobljenih dokumentov.

4 Zaključek

Velika količina digitalno shranjenih dokumentov v obliki nestrukturiranega besedila je povzročila potrebo po učinkovitem iskanju informacij. V tem članku smo obravnavali razvoj, temeljne koncepte ter sodobne pristope k iskanju informacij. Informacijsko iskanje se je skozi desetletja razvilo v kompleksno znanstveno področje, katerega jedro predstavljajo modeli za prodobivanje relevantnosti dokumentov glede na uporabniško poizvedbo. S klasičnih modelov, kot so Boolov model, vektorski model in probabilistični model, se je raziskovalna skupnost postopoma preusmerila k metodam s temeljem na strojnem učenju in globokih nevronskih mrežah.

Predstavili smo ključne značilnosti različnih modelov ter njihov razvoj skozi zgodovino, vključno z njihovimi prednostmi in slabostmi, ter opisali način, kako ti pristopi naslavljajo izzive naravnega jezika. Poseben poudarek smo namenili predstavitvi vgnezditev ter nevronskih modelov, ki omogočajo semantično razumevanje besedil ter učinkovito primerjavo dokumentov glede na vsebino.

References

[1] Inverted Index - GeeksforGeeks — geeksforgeeks.org. https://www.geeksforgeeks.org/inverted-index/, 2024. [Accessed 17-05-2025].

[2] Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. Docbert: Bert for document classification. *arXiv preprint arXiv:1904.08398*, 2019.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[4] Djoerd Hiemstra. Information retrieval models. *Information Retrieval: searching in the 21st Century*, pages 1–19, 2009.

[5] Ndengabaganizi Tonny James, Rajkumar Kannan, et al. A survey on information retrieval models, techniques and applications. *International Journal of Advanced Research in Computer Science and Software Engineering*, 7(7):16–19, 2017.

[6] KV Kanimozhi and M Venkatesan. Unstructured data analysis-a survey. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(3):223–225, 2015.

[7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[8] Bhaskar Mitra and Nick Craswell. Neural models for information retrieval. *arXiv preprint arXiv:1705.01509*, 2017.

[9] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[10] R Powar and B Arunkumar. Massive volume of unstructured data and storage space optimization-a review. *International Journal of Engineering & Technology*, 7(3.27):252–257, 2018.

[11] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

[12] Gerard Salton, Edward A Fox, and Harry Wu. Extended boolean information retrieval. *Communications of the ACM*, 26(11):1022–1036, 1983.

[13] Mark Sanderson and W Bruce Croft. The history of information retrieval research. *Proceedings of the IEEE*, 100(Special Centennial Issue):1444–1451, 2012.

[14] Alistair G Sutcliffe, Mark Ennis, and Jiawei Hu. Evaluating the effectiveness of visual user interfaces for information retrieval. *International Journal of Human-Computer Studies*, 53(5):741–763, 2000.

[15] Ayisha Tabassum and Rajendra R Patil. A survey on text pre-processing & feature extraction techniques in natural language processing. *International Research Journal of Engineering and Technology (IRJET)*, 7(06):4864–4867, 2020.

[16] Wikipedia. Vocabulary mismatch — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Vocabulary%20mismatcholdid=1267911918, 2025. [Online; accessed 25-May-2025].