

Hibrid következtetés a szövegfeldolgozásban

Hybrid inference in text processing

KILLÁN Imre

PTC Hungary Kft., 1066 Budapest, Nagymező u. 44., Tel: +36_20 6655951
ikilian@ptc.com

Abstract

Text processing by LLM-s has many advantages, but drawbacks too. It is not cheap, it has also great environmental footprint, but it is still weak in logical inference. The article proposes a hybrid solution to convert the text to knowledge graph by LLM technology, but for the following steps logical programming means are applied, grounded by ontologies. The article talks about question answering, logical inference and contradiction detection algorithms.

Keywords: LLM, text processing, hybrid solution, knowledge graph

Összefoglaló

Az LLM alapú szövegfeldolgozás előnyei mellett célszerű a hátrányairól is szólni: nem olcsó, nagy a környezeti lábnyoma, és különösen a logikai következtetések terén gyengécske. Erre javasol egy hibrid megoldást az alábbi cikk, amelyben a szöveget LLM technológiával tudásgráffá alakítjuk, további lépéseket azonban már különféle logikai programozási eszközökkel teszünk, ontológiák felhasználásával. Ennek körében következtetés, kérdés megválaszolás és ellentmondás kiszűrésének megoldását tárgyaljuk..

Kulcsszavak: LLM, szövegfeldolgozás, hibrid következtetés, tudásgráfok

1. BEVEZETÉS

Az LLM technológia térhódításának oka, de következménye is az a rengeteg újabb alkalmazási kérdés, amelyre sikeresen lehet őket alkalmazni. Egy ilyen terület a műszaki dokumentumok konzisztenciájának a vizsgálata. A különböző helyesírás-ellenőrző programok után, ha tartalmi relációkat, sőt, ha efféle hibákat, ellentmondásokat gépi eszközökkel fel tudunk fedeztetni, akkor mindenki számára csak növelt értékű, műszaki vagy egyéb dokumentációkat kaphatunk. Az LLM technológia azonban semmiképpen sem számít olcsónak – de tökéletesnek sem. Az alábbi cikkben a technológia szimbolikus következtetési módszerekkel való keverhetőségének lehetőségeiről lesz szó.

2. HIBRID KÖVETKEZTETÉSI TECHNOLÓGIÁK

Az [1] cikkben már irtunk hibrid következtetési technológiákról, amelyben egy képfeldolgozási technológia alkalmazásáról irtunk részletesen. Ebben egy YOLO algoritmust [3] kiegészítő szimbolikus következtetési fázis alkalmazásáról irtunk, amellyel nyilvános kamerákat egészítettünk ki intelligens következtetési lépésekkel, és ezáltal pl. otthagyt csomagokra figyelmeztettük a tömegközlekedés utasait.

Ugyanitt az LLM alapú és a hagyományos szabályalapú szövegfeldolgozási technológián alapuló csevegőprogramokat is összehasonlítottunk, és megállapítottuk, hogy a két alternatív megoldás más-más környezetben lehet jó – az LLM alapú megoldás szabad témájú kérdésekre gördülékeny fogalmazású, de sok esetben szószátyár válaszokat ad. Amikor valami kötött témában, csupán rögzített adatállományok lekérdezése és az eredmény szabatos és tömör mondatba foglalása a cél, akkor az LLM-ek szószátyársága egyenesen bosszantó is lehet.

A képfeldolgozásra alkalmas megoldás egy soros összekapcsolási módot követett, míg a két csevegőprogram választható alkalmazása párhuzamos megoldást mutatott be. Alább a tartalmi (szemantikus) viszonyokat elemző, nagyjából párhuzamos megoldásra mutatunk példát.

3. HIBRID SZÖVEGFELDOLGOZÁS

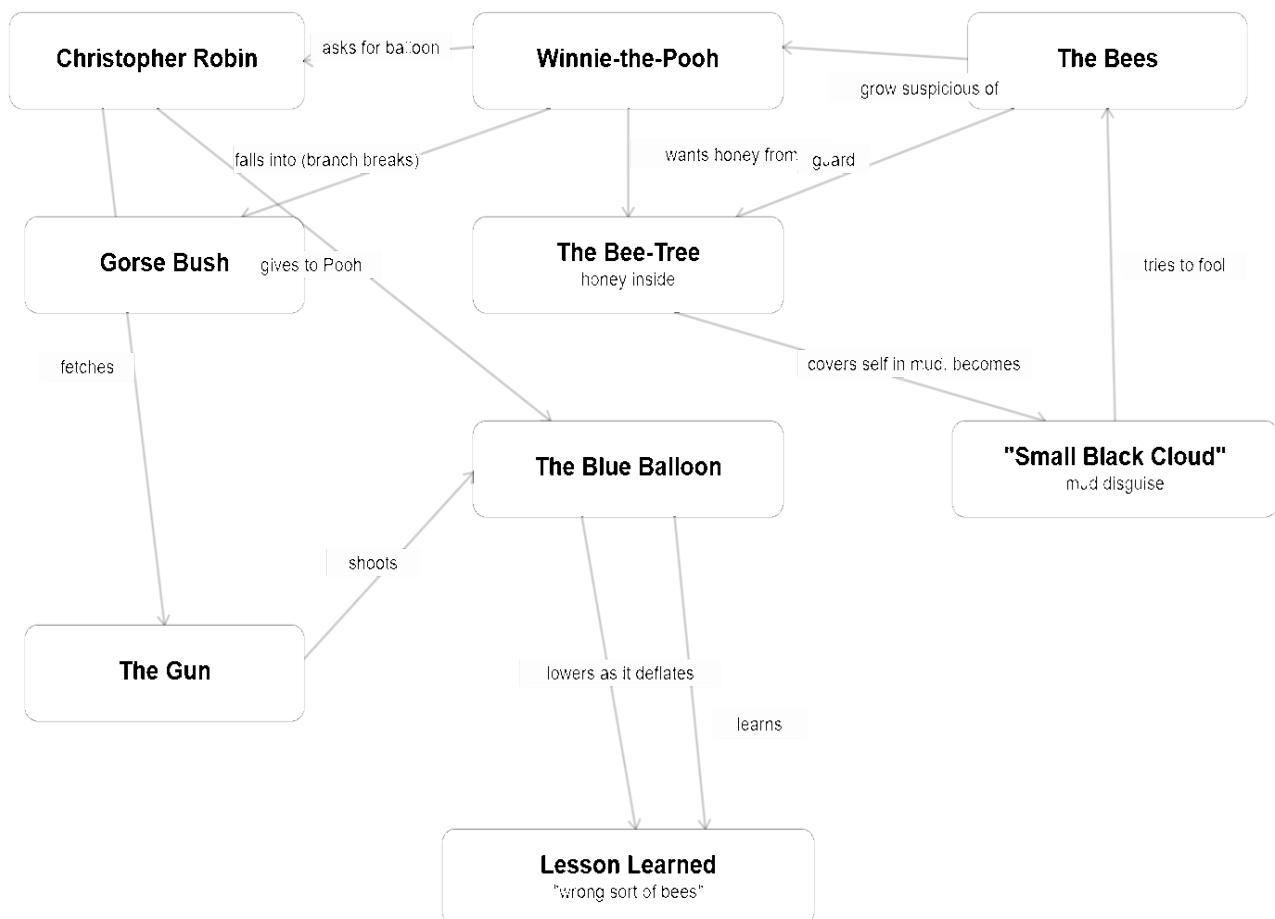
A szövegfeldolgozásra manapság használatos legkézenfekvőbb technológia az LLM alapú. A szabály-alapú megoldások még ha képesek is voltak bizonyos kötött nyelvi témában és korlátozott nyelvtannal kéréseket vagy mondatokat elemezni, a technológia általában csődöt mondott bármilyen új szó, különös, furcsa vagy akár hibás mondatszerkezet, netán hiányos szerkezetek feldolgozásakor és értelmezésekor.

Mindazonáltal az LLM alapú technológiák túl drágák. Hatásos méretű modellek már csak hatalmas szervertelepeken tudnak működni, amelyek egyrészt hálózati hozzáférést, másrészt olyan komoly tápellátást igényelnek, ami többek között már környezetvédelmi aggályokat is felvet.

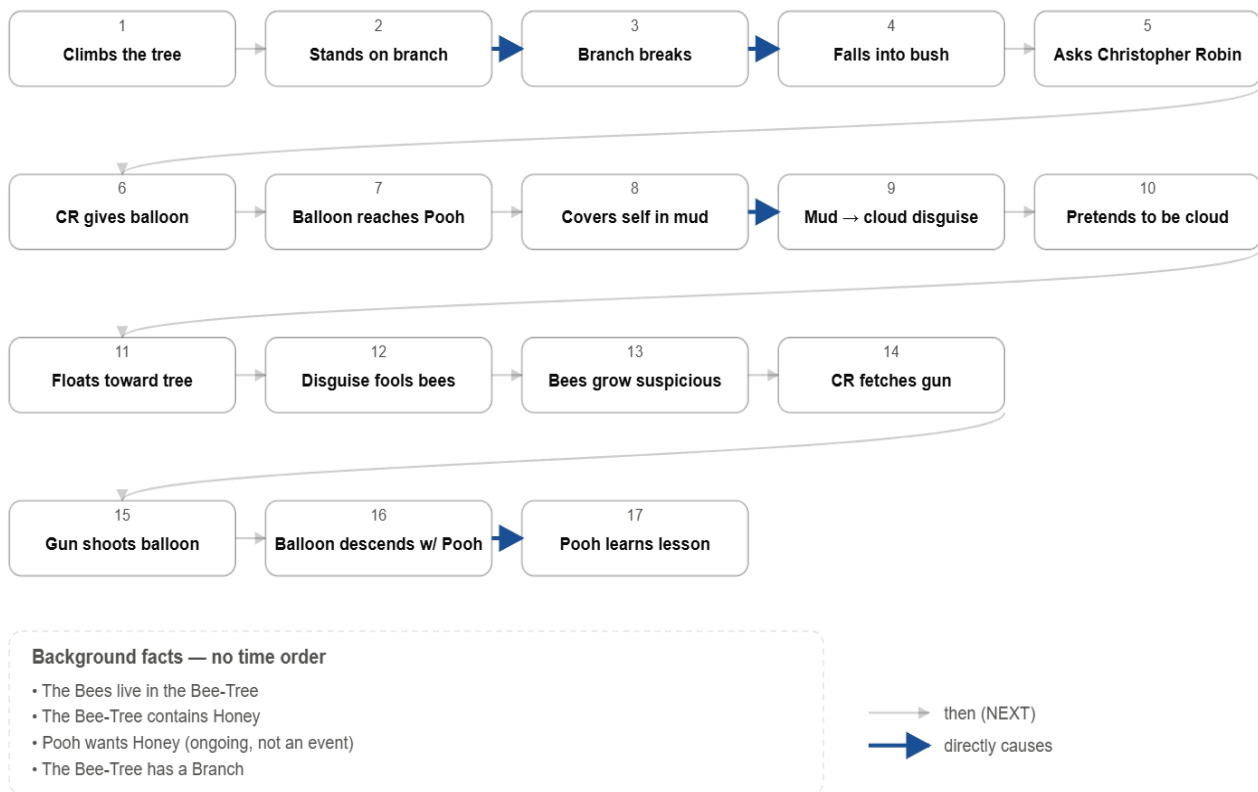
A jelzett gondokra megoldás lehet, ha a lényegesen olcsóbb, és helyi gépen is elvégezhető szimbolikus következtetési módszerrel próbálunk szemantikus műveleteket végezni. Ez inkrementális feldolgozás esetében eléggé kézenfekvő, vagyis amikor egy-egy újabb mondat vagy információegységen kívánunk logikai műveleteket végezni vagy konzisztenciát ellenőrizni, miközben a szöveg nagyobb részének már megvan a logikai leképezése.

3.1. Szövegek ábrázolása tudáshálóval

Közkezen forognak szövegeket a logikai tartalmukra leképező LLM alkalmazások, de komolyabb, ágens alapú alkalmazások (Copilot, Claude) csupán megfelelő promptlással is képesek efféle készíteni. Alább láthatjuk a Micimackó méhekről szóló fejezetének leképezését a Claude Code használatával, amely precízebb promptlással az alábbiánál még részletesebb ábrázolást is eredményezett.



Alább láthatjuk az időbeli és következmény-viszonyokra vonatkozó eredményt.



4. TUDÁSGRÁF ALGORITMUSOK

Mihelyt a szövegtartalom tudásgráf formájában létezik, alkalmas algoritmusok, sőt kész vagy még megvalósítandó szoftverek vannak, amelyek a legfontosabb funkciókra képesek. Megjegyzendő, hogy a szóban forgó következtetési rendszerektől általánosságban az alábbi tulajdonságokat várhatjuk el:

- Magyarázatadás. A rendszer legyen képes a kikövetkeztetett eredményt magyarázni is, vagyis leírni a következtetés lépéseit.
- Valószínűségi következtetés. Az említett működéshez elengedhetetlen egy valószínűségi változókon működő stratégia alkalmazása. Efféle kész rendszerek már léteznek, pl a Prolog nyelv CPLINT kiegészítése vagy a pythonra épülő ProbLog is efféle.
- Következtetési stratégia: következtetéshez a Horn-klózatokat – illetve annak Datalog részalmazát – használják, ahol a következményoldalon csak egyetlen állításelem állhat, és a logikában nincsen függvénykifejezés. Ebben a logikában a visszafelé haladó, vagyis egy célhipotézisből kiinduló stratégiára lényegileg a Prolog programnyelv a legjobb megvalósítás, ami viszont az előre haladó, vagyis a tényállításokból kiinduló stratégiát csak egyes kiegészítésekkel képes megvalósítani [6]. Valószínűségi következtetésekre sem képes alaptól – tegyük hozzá, ez nem is kézenfekvő kiegészítése a Prolog mechanizmusának [7].

A megvalósítandó funkciók:

- Ellentmondás mentesség (inkrementális értelemben, vagyis új információelemek ne mondjanak ellent egy nagyobb tudásgráf tartalmának)
- Következtetés a tudásgráfon (tényekből a szabályok segítségével új tények levezetése)
- Válaszadás (logikai változókat tartalmazó kérdésekre a változóértékeket, ill. a valószínűségi változókat leképező literálok mellett valószínűségi értékeket is kaphatunk)
- Szabálytanulás (tanítás, futtatás, kiértékelés)

4.1. Következtetés (entailment)

A következtetések logikai értelmezése kézenfekvő, a következménytények valószínűségének kiszámítása lehet kérdéses. Konjunkciókat tartalmazó szabály és független feltételek esetén a konjunkciós tényezők valószínűsége csak akkor szorozható össze, ha függetlenek egymástól, diszjunkciók esetében ugyanez a feltétele a valószínűségek összeadásának. A feladat egyáltalán nem egyértelmű, de konkrét megvalósítások léteznek rá [5],[6]. Mindkettő futhat Prolog alatt, a ProbLognak a python megvalósítása újabb keletű.

4.2. Válaszadás

A válaszadás a ProbLog Prolog alapú megvalósításában kézenfekvő: általánosságban visszafelé haladó (backward chaining) szabályalkalmazás esetén működik jól. Előre haladó esetben is megvalósítható, ilyenkor az előálló eredményhalmazt a kérdésre vonatkozólag szűrni kell.

A valószínűségi működésmód ezt némileg módosíthatja: itt ugyanis az eredményhalmaz minden egyes eleméhez egy valószínűségi érték is tartozik: a 0 érték jelzi a lehetetlen eseményt. A gyakorlatban gyakran alkalmaznak egy alsó határt, és ha bármilyen esemény valószínűsége ez alá süllyed, akkor azt kihagyják a számításból. Megjegyzendő, hogy a diszjunkciónál a független események kiértékelésekor összegződő alacsony valószínűségek még mindig kiadhatnak magas értéket, ezért az alsó határt semmiképpen nem célszerű túl magasra állítani, nehogy vele értékes eredményeket is kiszűrjünk.

4.3. Ellentmondás mentesség

Az ellentmondás mentességet az új információelemre vonatkozólag értelmezhetjük, amelyet valamiféle logikai literál, a célállítás képvisel. A következtetési képességet kihasználva az ellentmondást a célállítás negáltjára vonatkoztatva vizsgálhatjuk. Ha ez sikerre vezet, akkor a célállítás minden bizonnyal ellentmondásos.

4.4. Szabálytanulás

Szabály tanulásra az AnyBURL (Reinforced Anytime Bottom Up Rule Learning) algoritmus használható, amely Horn klózikra épül, ezért Prologra épülő, de tőle független megvalósításai is léteznek [8]. Az egész algoritmus lényege, hogy egy hatalmas ténybázison (tudásgráfon dolgozva) adott (nem túl nagy) hosszúságú relációs láncokat keres (pl. Imre apja István. István apja Endre.), majd keres egy közvetlen kapcsolatot a csomópontok között (pl. Imre nagyapja Endre). Ebből generálja a nagyapja(U,N):-apja(U,A),apja(A,N). szabályt. Erre vonatkozólag ellenőrzi, hogy a feltételszekvencia milyen változóértékeihez található meg a következmény-reláció is a ténybázisban, és ebből számol egy valószínűséget. Ha a valószínűség egy alsó korlát alá esik, akkor a szabályt eldobja.

Az AnyBURL algoritmus használati környezete hasonlít a neurális hálós technológiákhoz. Az elmondottakkal megtaníthatjuk a hálót bizonyos szabályokra.

A szabálytanítás után jön a futtatás: ilyenkor tetszőleges ténybázison futtatjuk a létrehozott valószínűségi szabályokat

Ha futtatni tudunk, akkor jöhet a validálás, amelyben konkrét ténybázisokon futtatjuk a szabályokat, ahol ismerjük az elvárt eredményeket, és ezeket összevetjük a kapott eredményekkel.

Amennyiben nagyobb adatállományaink is vannak, elvárt eredményekkel, (amelyek függetlenek a tanító adatoktól), akkor ezekre futtatva a validálást kiértékelést készíthetünk.

5. ONTOLÓGIÁK

Az ontológiák olyan logikai világmodellek, amelyek a való világ elemeit ábrázolják – egyrészt a világ elvonatkoztatott és konkrét fogalmait, másrészt azok egyedi példányait is, amennyiben azok bizonyos tudásszintnek megfelelnek. A fenti elképzelésekben az ontológiáknak két helyen is fontos szerepük van:

Egyrészt a tudásgráf az ontológia elemeihez kell, hogy legyen kötve: mind az egyes csomópont-egyedek révén, mind az őket összekötő relációk révén. Vagyis a csomópontok osztálypéldányok, a relációk pedig relációpéldányok kell, hogy legyenek.

Másrészt a leírt logikai következtetési műveletekben erősen építünk egyfajta háttértudásra – vagyis olyan szabályokra, amelyeket az ontológiák határoznak meg, és a hétköznapi tudásunkhoz tartoznak. Enélkül az említett következtetési lépések sokkal kevésbé lesznek hatékonyak, ill. az eredmények sokkal szegényesebbek lesznek.

6. A SZIMBOLIKUS KÖVETKEZTETÉS KORLÁTAI

A szimbolikus következtetési rendszerek alkalmazhatóságának komoly korlátai vannak.

- Az eldönthetetlenség – bár a Datalog maga még eldönthető, de a logikai reprezentáció bővítésével hamar beleütközhetünk az eldönthetetlenségbe
- Induktív következtetés: ha látunk 1000 fehér hattyút, kimondhatjuk-e, hogy minden hattyú fehér
- Fokozatok és szubjektivitás: hány hajszálnál kevesebb a kopasz?
- Elmosódó fogalmak, közhelyek, kulturális/helyzetfüggő környezet. Mikor mondjuk, hogy valaki „szép lány”, „derék ember” vagy „jó keresztény”?
- Nem monoton következtetés: „Brausewind egy madár. A madarak repülnek, tehát Brausewind is repül. Brausewind egy strucc, tehát mégsem repül. (Verne: Két évi vakáció)
- Helyzet/személy/kultúra/időfüggő következtetések, amelyek episztemikus/temporális logikával kezelhetők

7. TAPASZTALATOK

A fenti feladatok egyfajta hibrid technológiával megvalósíthatók. Erre azért van szükség, mert az LLM hívás nem olcsó, ezt próbáljuk minimalizálni. Másrészt az LLM technológia toleráns ugyan az ismeretlen szavakra, de nem eléggé megbízható – hallucinál, ha nem tudja a pontos választ, logikailag precíz következtetésre pedig nem is képes.

Mindazonáltal – bár épp a szerző jelentkezett az ITSy-Bitsy nevű szabályalapú nyelvi elemző programmal [2] – szövegek tudásgráffá alakításában csakis az LLM technológia jöhet számításba.

Ugy vélem, hogy szövegek, pl. műszaki dokumentumok esetében a szemantikus viszonyok, netán elentmondások felderítése egy alapvetően fontos és jól használható művelet. A fent leírtak mindenesetre egyelőre deszkamodell szintjén állnak, a termékszintre történő fejlesztés mindenképpen a jövő zenéje.

HIVATKOZÁSOK

- [1] Kilián Imre: Hibrid következtetőrendszerek In: SzámOkt 2022: XXXII Nemzetközi Számítástechnika és Oktatási Konferencia, Marosvásárhely, Erdélyi Magyar Műszaki Tudományos Társaság (EMT) 2022., pp 132-139.
- [2] Kilián Imre: Egy szabály- és következtetésvezérelt csevegőrobot modell anatómiája. Az ITSy-Bitsy modell. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, 2011. pp 265-274.
- [3] Juan Terven, Diana-Margarita Córdova-Esparza, Julio-Alejandro Romero-González: A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS, Machine Learning & Knowledge Extraction Nov-2023. (<https://www.mdpi.com/2504-4990/5/4/83>, elérés: 2026-Szept-23)
- [4] Damiano Azzolini, Matteo Bonato, Elisabetta Gentili, Fabrizio Riguzzi: Logic Programming for Knowledge Graph Completion (<https://ceur-ws.org/Vol-3733/paper4.pdf>, elérés 2026-Szept-23)
- [5] Machine Learning Group, University of Ferrara: CPLINT Tutorial (<http://cplint.lamping.unife.it/tutorial/tutorial.swinb> elérés: 2026-Szept-23)
- [6] DTAI Group, TU Leuven: ProbLog ([Introduction.](#) — [ProbLog: Probabilistic Programming](#), elérés, 2026-Szept-23)
- [7] Robin Manhaeve, Sebastijan Dumančić, Angelika Kimmig, [Thomas Demeester](#), [Luc De Raedt](#): Neural probabilistic logic programming in DeepProbLog, In: Artificial Intelligence, Volume 298, 2021, (<https://www.sciencedirect.com/science/article/pii/S0004370221000552>, elérés: 2026-Szept-23)
- [8] Meilicke, C., Chekol, M.W., Betz, P. *et al.* Anytime bottom-up rule learning for large-scale knowledge graph completion. In *The VLDB Journal* **33**, pp. 131–161 (2024) (<https://doi.org/10.1007/s00778-023-00800-5>, elérés 2026-Szept-23).