NEMZETBIZTONSÁGI SZEMLE
13. évfolyam (2025) 1. szám 77–95. | DOI: 10.32561/nsz.2025.1.6

Bottyán Sándor[1]

A nagy nyelvi modellek működése és képzése, valamint alkalmazásuk stratégiai elemzése

1. rész

The Operation and Training of Large Language Models and a Strategic Analysis of Their Application

Part I.

Napjainkban egyre több olyan munkakörnyezeti szoftvermegoldás jelenik meg, amelyeket mesterségesintelligencia-ügynökök támogatnak, és amelyek fő képességeit nagyméretű nyelvi modellek biztosítják. E modellek megfigyelhető fejlődési irányai (például erőforrás-fogyasztás csökkentése, kompakt architektúrák) olyan jövőt sejtetnek, amelyben ez a technológia valószínűleg mindennapi eszköztárunk részévé válik az információfeldolgozásban, beleértve a nemzetbiztonsági feladatokat is. A hatékony és felelősségteljes használathoz meg kell érteni e technológia alapelveit és fejlesztési szükségleteit, valamint fel kell tárni a gyakorlati alkalmazásban rejlő pozitív és negatív tulajdonságait. Az ismeretterjesztő célú kétrészes tanulmány első részében a mesterséges intelligencia és a gépi tanulás tágabb területén a transzformátormodellekre és azok leglényegesebb technológiai alapjaira összpontosítunk, hogy demisztifikáljuk ezt a gyakran homályosnak és nehezen érthetőnek tartott jelenséget, és közelebb hozzuk a társadalmi elfogadottsághoz.

Kulcsszavak: *nagy nyelvi modell, mesterséges intelligencia, ChatGPT, gépi tanulás*

Nowadays, an increasing number of work environment software solutions are emerging that are supported by artificial intelligence agents, where the main capabilities are provided by large language models. The observed development directions of

[1] Hallgató, Nemzeti Közszolgálati Egyetem Hadtudományi és Honvédtisztképző Kar Katonai Műszaki Doktori Iskola, e-mail: bottyan.sandor@gmail.com

these models (e.g., reducing resource consumption, compact architectures) suggest a future in which this technology will likely become part of our everyday toolkit in information processing, including national security tasks. Effective and responsible use requires understanding the fundamental principles and development needs of this technology, as well as uncovering both its positive and negative features in real word application. In the first part of this two-part study aimed at knowledge dissemination, we focus on transformers and their most essential technological fundamentals in the broader fields of artificial intelligence and machine learning, in order to demystify this phenomenon – often regarded as obscure and difficult to understand – and bring it closer to societal acceptance.

Keywords: *large language model, artificial intelligence, chatGPT, machine learning*

Bevezetés

Az elmúlt évtizedekben a digitalizáció hatása hazánk modern társadalmának számos aspektusában növelte az elektronikusan tárolt információk mennyiségét, egyre több elektronikus információs rendszer végzi az adatok létrehozását, kezelését, továbbítását a kibertérben. Ez a jelenség az állambiztonságot garantáló szervezetek tevékenységeire is hatással van, a nemzetbiztonsági intézményrendszerre egyre nagyobb teher hárul az elektronikus információk kezelése kapcsán. A hírszerzést végző nemzetbiztonsági szervek számára jelentős kihívás az alaptevékenységük során megszerzett adatok mennyiségének elemzése és értékelése, az időszerű, értéket képviselő információ előállítása, ezért a hagyományos hírszerzési eljárásokat manapság már új, automatizált információfeldolgozó képességekkel vértezik fel. Ez a folyamat napjainkban is tart, az új integrálandó képességek közül a természetes nyelvfeldolgozás (*natural language process,* NLP), valamint a népszerű mesterséges intelligencia (MI) technológiák, mint a gépi tanulási megoldások egyre több állami és katonai feladatot ellátó eljárásban jelennek meg. Az ilyen innovatív technológiák integrálásának szükségét vagy az önálló fejlesztések indokoltságát az olyan nagy nyelvi modellek (*large language models,* LLM) képességei támasztják alá, mint például a ChatGPT, a Gemini vagy a Claude. Az LLM alkalmazásában rejlő előnyök (például automatizáltság, döntéshozatal-támogatás, hatékonyságnövelés stb.) hatékonyan hasznosíthatók a katonai és a nemzetbiztonsági információfeldolgozás szintjein, amelynek köszönhetően a hatalmas adatkészletek automatizált elemzése gyorsítja a folyamatokat. Az előre definiált minták felhasználásával az adathalmazokban azonosíthatók az állambiztonságot sértő új fenyegetések (például Magyarország függetlenségére, gazdaságának biztonságára, pénzügyi helyzetére veszélyes külföldi szándékok és cselekmények stb.) és a már ismert kockázatok állapotában történt – soron kívüli intézkedést igénylő – változások. Összefüggések felismerésével prediktív és preskriptív előrejelzések készülhetnek, amelyek hozzájárulnak a társadalomra veszélyes cselekmények kockázatainak csökkentéséhez vagy elkerüléséhez, a nemzetbiztonsági védelem alá eső szervezetek és létesítmények védelméhez. A technológia skálázási lehetősége megengedi a vizsgált adatok mennyiségi kiterjesztését, amellyel még hatékonyabbá válhat a magyar nemzetbiztonságot veszélyeztető külföldi tevékenységekkel (például terrorszervezetek tervezett cselekményei,

a kábítószer- és fegyverkereskedelem trendjei stb.) kapcsolatos információgyűjtést követő elemzés-értékelési munka. A nyílt forráskódú *(open source)* nagy nyelvi modellek kiválóan alkalmasak az ilyen tevékenységek támogatására, minden kétséget kizáróan ezen innovációs kezdeményezéseknek otthon adó területek közül a hazai nemzetbiztonsági terület sem lesz kivétel. Az egyedi ágens fejlesztések mellett egyre gyakoribb jelenség az LLM felhasználása asszisztencia jellegű (például tartalomgenerálás, fordítás, kódolás) szolgáltatásokban is, több technológiai nagyvállalat integrálja azt munkakörnyezeti szoftverekbe (például Microsoft Office 365 – Copilot, Google Workspace – Assistant) is. Bizonyos, hogy a technológia a katonai és nemzetbiztonsági munkakörnyezetek általános eszköztárának a része lesz, így a biztonságtudatos fejlesztés és használat előmozdítása érdekében elengedhetetlen az alapvető működés megértése és az ezzel kapcsolatos jellemzők (előnyök, lehetőségek, fenyegetések, veszélyek) azonosítása. A technológia alkalmazása többrétű biztonsági kihívással jár együtt, ami részben *black box*[2] jellegéből adódik, másrészről abból, hogy bonyolult működési struktúrája okán nehezen értelmezhető. Komplexitása kihívást jelent a nem szakértő személyek számára, ennek okán a vele való munkavégzés sem problémáktól mentes. Ez a kétrészes tanulmány ismeretterjesztő céllal demisztifikálni kívánja a nagy nyelvi modellek jelenségét, átfogó képet kíván nyújtani az alapvető működésről és szükségletekről, mindemellett rávilágít az alkalmazásában rejlő kockázatokra is. Arra a várható jövőképre alapozva, hogy az LLM-kockázatok hamarosan az állami szervezetek szintjén fognak jelentkezni, összeállított ismeretanyagával és következtetéseivel igyekszik hozzájárulni a felelősségteljes alkalmazáshoz, így növelve a nemzetbiztonság szempontjából igen fontos elektronikus információbiztonságot, valamint a mesterséges intelligenciába vetett társadalmi bizalmat.

A mesterséges intelligencia fogalmi megközelítése

A mesterséges intelligencia (MI), vagy angolul *artificial intelligence (AI)* mostanság gyakorta ismételt fogalom, amely hosszú történeti múltra tekint vissza. Az első MI-eredményt 1943-ra jegyzik, ekkor történt, hogy Warren McCulloch és Walter Pitts az agyi neuronokról szóló alapszintű fiziológiai ismereteikhez társították a matematikai elemzés módszertanát, így megalkotva a legelső mesterséges neuronmodellt, amely nélkül ma nem beszélhetnénk korszerű MI-megoldásokról. A következő években sorra jelentek meg olyan kutatások és szabályok (például Hebb-tanulás 1949-ben) vagy műszaki megoldások (például Marvin Minsky és Dean Edmonds SNARC elnevezésű neurális számítógépe 1951-ben), amelyek leginkább ezekre az ismeretekre alapoztak. Azonban talán a legteljesebb elképzelés Alan Turinghoz köthető, akinek 1950-ben megjelent megfogalmazásai (például Turing-teszt, a gépi tanulás, megerősített tanulás) manapság is gyakran említett kifejezésnek számítanak az MI-terminológiában.[3]

[2] A *black box* kifejezés a technológia és szoftverek esetében olyan rendszert vagy komponenst jelöl, amelynek belső működése a felhasználó vagy a megfigyelő számára nem átlátható vagy hozzáférhető. Csak a bemeneti adatok és a kimeneti eredmények ismertek, de maga a folyamat, ahogyan a rendszer a bemenetet a kimenetre alakítja, rejtve marad.

[3] Bottyán 2024: 31.

A mesterséges intelligencia története során több MI-definíciót is alkottak, de a technológia rohamos fejlődése okán azok előbb-utóbb elavulttá váltak, így nincs egy egységesen elfogadott közülük. Az MI-definíciós kísérletek három perspektívából alkothatnak maradandó gondolatot, ezek a filozófiai, a technológiai és a jogi, amelyek közül a technológiai megközelítések nem tekinthetők konzisztensnek, hiszen a gyorsvonatként száguldó innovatív technológiát pontos műszaki paraméterekkel időtállón megfogalmazni lehetetlen küldetés. Azonban a filozófiai megközelítések permanensnek bizonyultak, és napjainkban népszerűnek is mondhatók. Például John. R. Searle 30 éves definíciója, aki az MI magyarázatát a kognitív állapot elérése mentén kétfelé, erős és gyenge MI-re osztotta.[4] Ez a megközelítés, bár gyakorta alkalmazott az általános megközelítések során, de a jelenkori MI-rendszerek – vizsgálati szempontból – nem klasszifikálhatók e rend szerint, hiszen minden általunk ismert megoldás a gyenge MI követelményeinek felel meg, míg az erős MI jelenleg csak vízióként jelenik meg. Bár a szabályozás mindig is utánkövetője volt az innovatív technológiáknak, napjainkban már rendelkezünk a mesterséges intelligencia egy olyan jogi deklarációjával, amely a következő időszakokra megadhatja azt a pontosságot, egyértelműséget és interdiszciplinaritást, amelyre a szükség kívánkozik. Az európai kezdeményezés [2024/1689 EU rendelet][5] a következőképpen határozza meg a mesterséges intelligenciát:

> „»MI-rendszer«: gépi alapú rendszer, amelyet különböző autonómiaszinteken történő működésre terveztek, és amely a bevezetését követően alkalmazkodóképességet tanúsíthat, és amely a kapott bemenetből – explicit vagy implicit célok érdekében – kikövetkezteti, miként generáljon olyan kimeneteket, mint például előrejelzéseket, tartalmakat, ajánlásokat vagy döntéseket, amelyek befolyásolhatják a fizikai vagy a virtuális környezetet."

A jogalkotó feladata szerint igyekezett az általunk tapasztalt mesterségesintelligencia-valóságot az írott természetes nyelv határai között leképezni. A fentebb megfogalmazott gondolatokkal a jog erejének (kötelező alkalmazás, a jogalkalmazás alapja, jogbiztonság garantálása) köszönhetően jó eséllyel egyre gyakrabban fogunk találkozni. A fenti definícióból következtetve, ez a tanulmány a mesterséges intelligencia tartalomgenerálási képességeinek főbb részleteit járja körbe. Ehhez a gépi tanulás területén keresztül kívánja megismertetni az olvasót a nagy nyelvi modellekkel.

A gépi tanulás

A gépi tanulás jellemzésére Arthur Samuel gondolatai szolgálhatnak segítségül, aki szerint ez a számítástechnika azon részterülete, amely anélkül képes tanulni, hogy programoznánk azt.[6] Ezek az algoritmusok statikus programutasításokat követve a jókora adattömegekből

[4] Bottyán 2024: 32.

[5] Az Európai Parlament és a Tanács (EU) 2024/1689 rendelete (2024. június 13.) a mesterséges intelligenciára vonatkozó harmonizált szabályok megállapításáról, valamint a 300/2008/EK, a 167/2013/EU, a 168/2013/EU, az (EU) 2018/858, az (EU) 2018/1139 és az (EU) 2019/2144 rendelet, továbbá a 2014/90/EU, az (EU) 2016/797 és az (EU) 2020/1828 irányelv módosításáról (a mesterséges intelligenciáról szóló rendelet).

[6] Munoz 2012.

(bemeneti minta) képesek modellt építeni (tanulni) és előrejelzéseket (predikciókat) készíteni. A gépi tanulás szoros kapcsolatban áll a számítógépes statisztikával, és gyakran átfedésbe kerül az adatbányászat módszertanával. Az adatelemzés területén előjelzések céljából alkalmazott gépi tanulási módszereket pedig *prediktív analitikának* nevezik. A gépi tanulás kategóriáján belül két elterjedt alkalmazási módszerről beszélhetünk, amelyek a *felügyelt és a felügyelet nélküli tanulás*. A megjelenés gyakorisága tekintetében ezek közül előbbi a jelentősebb (körülbelül 70%), míg utóbbi 10–20% arányban jelenik meg, ugyanakkor meg kell említeni a *félig felügyelt* és a *megerősített tanulás* módozatokat is, amelyek azonban ritkábban alkalmazott módszerek.[7]

A felügyelt tanulás során címkézett példák segítségével képzik ki a modellt úgy, hogy a kimenet előre definiált. Ebben az esetben az algoritmus úgy tanul, hogy a két készletet (bemenet és kimenet) összeveti, és megalkotja a modellt annak érdekében, hogy a címkézetlen adatok tekintetében is hatékony legyen. Ehhez leginkább olyan módszereket alkalmaz, mint például a *regresszió,*[8] a *predikció,* a *gradient boosting.*[9] A felügyelt tanulást leginkább olyan célokra használják, ahol a múltbéli adatok feldolgozásával valószínűsíthetők bizonyos jövőbeli események.[10]

A felügyelet nélküli tanulást akkor alkalmazzák, amikor az algoritmusnak egy *strukturálatlan* adattömegben kell mintákat megtalálnia. Címkézett adatok nélkül a cél az adatokban található egyfajta *szabályosság* megállapítása. Ez a módszer különösen hatékonyan alkalmazható pénzügyi tranzakciós adatok elemzésére, hiszen hasonló jellemzőkkel rendelkező ügyfelek csoportjai azonosíthatók (klasszifikálhatók), majd a kapott eredmények felhasználhatók további szervezeti célok (például marketingtevékenység) érdekében. Az alkalmazott módszerek az *önszerveződő térképek,*[11] *a legközelebbi szomszéd analízis,*[12] *K-közép klaszterezés*[13] és a *szinguláris felbontás*.[14]

A félig felügyelt tanulást *(semi-supervised learning)* többnyire ugyanazon tevékenységekre alkalmazzák, mint a felügyelt tanulást, azonban a modellképzéshez címkézett adatokat csak kis mértékben, míg a címkézetlen adatokból nagy mennyiségben használnak fel. Ez a módszer annyiban tér el a felügyelt tanulástól, hogy a fejlesztési folyamat gazdaságosabbá válhat abban az esetben, ha az adatok címkézése tetemes költségekkel járna.[15]

---

7 Ongsulee 2017: 2.

8 A regresszió statisztikai módszert arra használják, hogy megvizsgálják a változók közötti kapcsolatokat.

9 A gradiens erősítés gépi tanulási technika, amely a döntési fák sorozatát használja előrejelzések készítésére és a prediktív teljesítmény javítására.

10 Ongsulee 2017: 2.

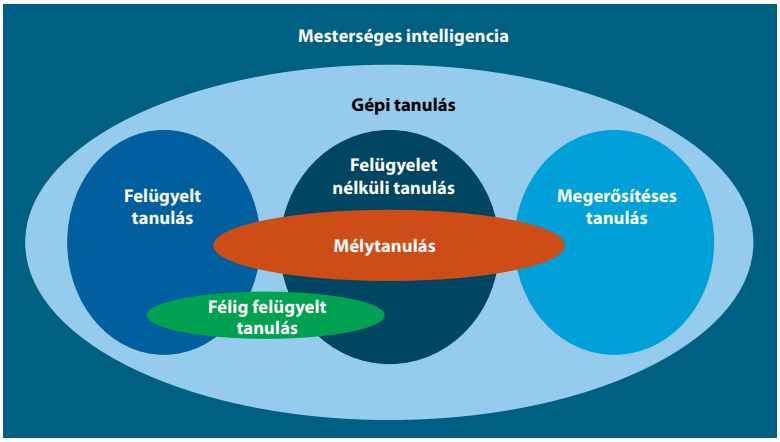
11 Az önszerveződő térkép *(self-organizing map)*, más néven Kohonen-térkép, egyfajta mesterséges neurális hálózati technika, amelyet nagydimenziós adatok alacsonyabb dimenziókban történő feltérképezésére és megjelenítésére használnak.

12 A legközelebbi szomszéd analízis a térbeli rendezettség vizsgálatának gyakorta alkalmazott matematikai statisztikai eszköze.

13 A k-közép klaszterezés *(K-means clustering)* egy népszerű, nem felügyelt tanulási algoritmus, amelyet adathalmazok klaszterekbe történő csoportosítására használnak. A cél az, hogy az adatok egy adott számú (K) klaszterbe legyenek osztva úgy, hogy az egy klaszterbe tartozó elemek hasonlók legyenek egymáshoz, míg a különböző klaszterek elemei minél jobban különbözzenek.

14 A szinguláris érték-felbontás *(SVD – singular value decomposition)* egy lineáris algebrai módszer, amely egy mátrixot három másik mátrix szorzatára bont fel. Az SVD széles körben alkalmazott eszköz adatelemzésben, képfeldolgozásban, gépi tanulásban és kompresszióban.

15 Ongsulee 2017: 3.



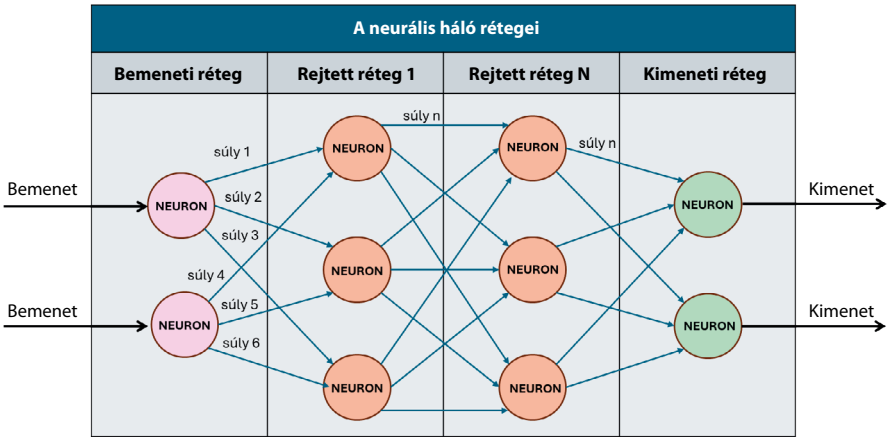
1. ábra: A mesterséges intelligencia tanulástípusainak szemléltetése hierarchiájuk és kapcsolataik alapján
Forrás: a szerző szerkesztése

A megerősített tanulás *(reinforcement learning)* szintén önálló kategória, amely megoldás leginkább a robotikában és a játékokban (például sakk) hasznosul. Ebben az esetben az algoritmus kísérletezéssel fedezi fel azt, hogy melyik művelet eredményei adják a legnagyobb jutalmat, vagyis a megerősítést. A módszernek három összetevője van: az *ágens* (a döntéshozó), a *környezet* (amivel az ágens kapcsolatba léphet) és az *akciók* (amit az ágens megtehet). A cél ebben a tekintetben az, hogy az ágens olyan döntéseket hozzon, amelyek maximalizálják a várható jutalmat.[16]

A mélytanulás *(deep learning),* vagy más néven mély gépi tanulás, egy speciális gépi tanulás, amely nem azonos az eddigiekben említett tanulási paradigmákkal, de alkalmazható azokban (1. ábra). Jellegzetessége, hogy egynél több mesterséges neurális hálózatot (úgynevezett mélyhálót) és hozzá kapcsolódóan gépi tanulási algoritmusokat (például tanulóalgoritmusokat) tartalmaz. A mélytanulás elnevezés neuronhálózatban található több rejtett réteg meglétére utal. A mesterséges neuronok alkalmazása nem új technológia, ám a jelenkor számításikapacitás-lehetőségeivel és egyéb algoritmikákkal társítva dinamikusan fejlődő terület.[17]

Egy neuronhálózat legegyszerűbb esetében két neuroncsoport megléte szükséges, amelyből az egyik bemeneti, a másik pedig kimeneti réteg. Egy komplex mélyhálóban a bemenet és kimenet között további nagyszámú neuron található, amelyek szintén régetekbe (úgynevezett rejtett rétegekbe) tömörülnek (2. ábra). E rétegekben található mesterséges neuronok bemeneti jeleket (valós számokat) fogadnak a vele kapcsolatban álló neuronoktól, majd azt egy aktivációs függvénnyel[18] feldolgozzák, ezt követően továbbítják. A jelek a teljes hálózaton áthaladnak *(forward propagation)*, amely átalakítja azokat a kívánt módon. A jel erősségét, avagy a neuronok közötti kapcsolatot minden csatlakozásnál egy súly *(weight)*

16 Ongsulee 2017: 3.
17 Bottyán 2024: 35.
18 Az aktivációs függvény *(activation function)* egy matematikai függvény, amelyet a neurális hálózatokban a neuronok kimenetének meghatározására használnak. Ez a függvény dönt arról, hogy egy neuron aktiválódik-e, és milyen mértékben járul hozzá a következő réteg bemenetéhez.



2. ábra: A klasszikus (fully connected) mélytanulási mesterséges neuronháló rétegtípusainak szemléltetése
Forrás: a szerző szerkesztése Malik *2019 ábrája alapján*

határozza meg, amelyek kezdetben véletlenszerű értékeket kapnak, de a tanítási folyamatban a visszacsatolás során *(backpropagation)* kalibrálódnak (súlyozódnak).

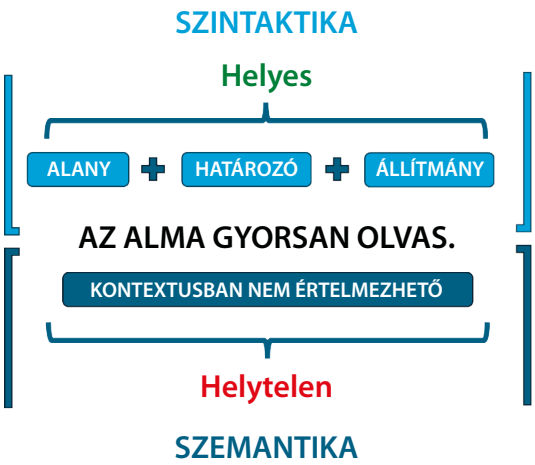
Az agyi neuronok működéséhez hasonlóan a mesterséges neuronok is az előre kalibrált feltételek (ingerek) fennállása esetén aktivizálódnak (tüzelnek). Az aktiváció sikerességéhez járul hozzá a torzítás vagy eltolás *(bias),* amelyek a bemenetekhez adódnak hozzá (egyfajta „súlyként") azért, hogy amennyiben egy bemenet összege önmagában nem lenne elegendő az aktiváláshoz, a neuron akkor is aktivizálódjon.[19] A mélyhálót kezdetekben egyszerű klasszifikációra alkalmazták, azonban eltérő típusú feladatokban is érvényesülnek, a nyelvi feldolgozás során úgynevezett *rekurrens* (visszacsatolt) neuronhálókat alkalmaznak annak érdekében, hogy a minták közötti kapcsolatot is reprezentálják.[20]

Természetesnyelv-feldolgozás

A természetesnyelv-feldolgozás (*natural language processing,* NLP) tudományos és technológiaalapú kutatási terület. A nyelvészet, az informatika és a mesterséges intelligencia egyik olyan részterülete, amely az emberi nyelvek reprezentációjának és automatikus elemzésének számítási technikáit foglalja magában. Az emberi nyelvek szövegszerű elemzése, annak kellő mértékű megértése még a gépek számára sem egyszerű feladat, hiszen abban számos nehezítő tényező (például nagy mennyiségű megjelenés, összetettség, többértelműség stb.) merül fel. Az automatizált NLP célja, hogy egy adott időkereten

[19] Intuitív példaként képzeljünk el egy neuronhálót, amelynek feladata macskák felismerése képeken. Alapesetben a hálózat csak akkor tudja felismerni a macskát, ha minden vonás (fülek, szemek, bajusz) tökéletesen illeszkedik. A *bias* lehetővé teszi, hogy kisebb eltérések esetén is felismerje a mintát.

[20] Az az eljárás, amely során a nyers adatokat olyan formátumba alakítjuk át, amelyet a mély neurális hálózatok hatékonyan tudnak feldolgozni. Ongsulee 2017: 2.



3. ábra: Egy mondat lehet szintaktikailag helyes, miközben kontextusbeli jelentése értelmetlen (egy alma nem képes „gyorsan olvasni"), vagyis szemantikailag hibás
Forrás: a szerző szerkesztése

belül a nyelvi szövegek bőséges mennyiségében rejlő nagy tudástartalmat, bölcsességet hatékonyan és nagy pontossággal felfedezze.[21]

Az NLP alkalmazási területei többnyire az alábbiak:

- automatikus nyelvi fordítás,
- információkeresés,
- információ kinyerése,
- kérdések és válaszolás,
- nagy méretű szövegek indexelése és keresése,
- szövegek automatikus összefoglalása,
- szövegek klasszifikációja,
- szöveggenerálás/dialóguselemzés.[22]

Az NLP legegyszerűbb megközelítései a *szintaktikai és a szemantikai elemzés* módszertana. A szintaktikai elemzés során, a nyelvtan alapján meghatározzák a vizsgált mondat szerkezetét, valamint az igék alanyát és tárgyát (3. ábra). Ha a szerkezet meghatározása megfelelő, akkor további operátor-operandus kapcsolatok meghatározásával érkezünk el a szintaktikai elemzés kimeneteléhez.[23] A szemantikai elemzés célja annak a meghatározása, hogy mit jelent és mit tartalmaz az adott mondat, túllépve a szavak önálló értelmezésének szintjén. Leginkább arra törekszik, hogy megértse a kifejezések és a mondatok mögötti kontextusbeli kapcsolatokat és a rejtett információkat.[24]

A szemantikai elemzés a kérdő és felszólító mondatok esetében körülményesebb, ugyanakkor az olyan feladatokban, mint a gépi fordítás, az összefoglaló-készítés,

[21] Chowdhary 2020: 605–608.
[22] Chowdhary 2020: 608–609.
[23] Chowdhary 2020: 608–610.
[24] Bottyán 2024: 37.

a kérdésmegválaszoló rendszerek és az érzelmek elemzése kulcsfontosságú megközelítés.[25] A felsoroltakon kívül említhetnénk még elemzési módozatokat (például pragmatikai, diskurzus, szövegkohéziós stb.), de mivel nem célunk tárgyalni azokat, ettől eltekintünk.

A nagy nyelvi modellek

Az NLP történeti fejlődése átível a statisztikai nyelvi modellezésen, a neurális nyelvi modellezésen, az előre tanított nyelvi modelleken *(pretrained language model,* PLM) át a nagy népszerűségnek örvendő nagy nyelvi modellekig. A hagyományos nyelvi modellek *(language model)* eleinte csak feladat-specifikációra fejlesztett felügyelt rendszerek voltak, az őket követő PLM-modelleket már önfelügyelt[26] környezetben, nagy szövegkorpuszokon[27] tanították.[28] A nagyobb PLM-ek egyre több paramétert[29] tartalmaztak elődjeiknél, és természetesen a képzésükhöz felhasznált adatállomány mérete is jelentősen nőtt. Ennek köszönhetően a – koherens kommunikáció mellett többféle feladatra is alkalmazhatóvá váló – modellek teljesítménye is tovább növekedett, és egy új elnevezéssel már nagy nyelvi modellekként hivatkozunk rájuk.[30] Amennyiben néhány sorban kellene konzisztensen tisztázni a nagy nyelvi modellek lényegét, arra az alábbiak szerint tesz a szerző definíciós kísérletet:

A nagy nyelvi modellek olyan jelentős mennyiségű képzési adaton tanított mesterségesintelligencia-rendszerek, amelyek architektúrájukban a mélytanulást alkalmazva képesek bizonyos típusú nyelvi feladatok széles skáláját nagy pontossággal ellátni.

A nagy nyelvi modelleket architektúrájuk alapján az alábbiak szerint csoportosíthatjuk (4. ábra):

- A kódoló *(encoder)* típusú architektúra esetében a bemeneti adat fix hosszúságú vektorként reprezentálódik, ami ezt követően más feladatok bemeneteként használódik fel. Az ide sorolható LLM-ek többnyire csak a nyelv megértésére alkalmasak, és olyan funkciókat látnak el, mint például a klasszifikáció vagy a szentimentelemzés.[31]
- A dekódoló *(decoder)* típusban rejtettállapot-vektorból indul ki a feldolgozás, és úgy generálódik a kimenet lépésről lépésre, hogy figyelembe veszi a bemeneti, valamint az addig generált adatokat egyaránt. Rendkívül eredményes olyan generatív képességekben, mint a szöveg-előállítás, történetírás vagy válaszok generálása.

25 Chowdhary 2020: 606.

26 Az önfelügyelt tanulás során a modell olyan feladatokat kap, amelyek a meglévő adatokból generálhatók.

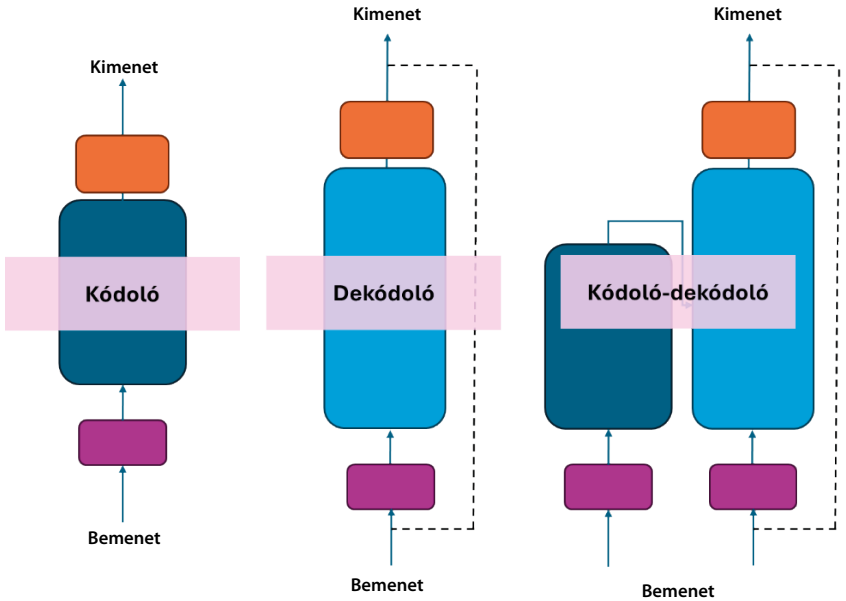
27 A „korpusz" a nyelvészetben és a szöveges adatokkal dolgozó területeken egy nagy, strukturált szöveggyűjteményt jelent, amelyet tudományos kutatásokhoz, nyelvi modellek betanításához vagy egyéb nyelvi feladatokhoz használnak.

28 Devlin et al. 2018.

29 A paraméterek a modellben lévő súlyokat jelentik, amelyek a tanulási folyamat során állítódnak be annak érdekében, hogy a bemeneti szöveg alapján pontos kimenet generálódjon. A nagy nyelvi modellek (LLM-ek) esetében a paraméterek nem maguk a mesterséges neurális hálózat neuronjai, hanem inkább a hálózat azon súlyai és eltolásai (angolul: *weights and biases*), amelyek a neuronok közötti kapcsolatokat határozzák meg.

30 Raffel et al. 2020: 5; Agüera y Arcas 2022.

31 A szentimentelemzés *(sentiment analysis)* egy olyan NLP-technika, amely szövegek érzelmi tartalmát elemzi és kategorizálja.



4. ábra: A nagy nyelvi modellek architektúratípusainak ábrázolása
Forrás: a szerző szerkesztése Baswal *2023 alapján*

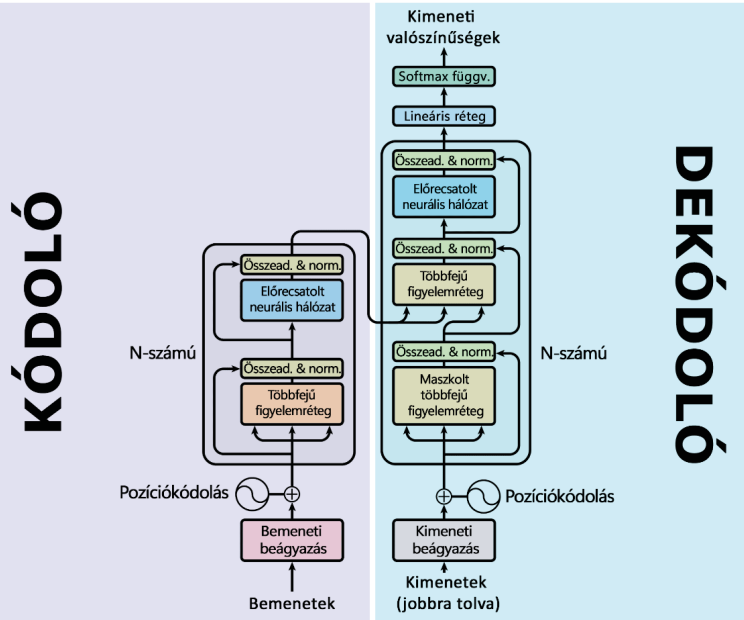
- A kódoló-dekódoló *(encoder-decoder)* architektúrák egyaránt egyesítik mindkét típus összetevőit. Kiválóak olyan feladatokban, ahol a bemeneti és kimeneti tartalom között szoros kapcsolat van, mint például a fordítás vagy a szövegek összefoglalása.

A kódoló-dekódoló csoportba sorolható a transzformátor- vagy transzformátorhálózat-architektúratípus is, amely az automatikus természetesnyelv-feldolgozásban a forradalmat – a nagy nyelvi modellek eredményeiben pedig az áttörést – hozta el.[32] A következőkben kifejezetten ennek sajátosságait, valamint a modellek mögött alkalmazott eljárásokat és technikákat vesszük górcső alá.

A transzformátorarchitektúra

A mesterséges intelligencia napjainkban tapasztalt társadalmi népszerűsége leginkább az OpenAI vállalat transzformátoralapú ChatGPT szolgáltatásával azonosul, ennek ellenére az architektúra működését először a Google szakemberei mutatták be az *Attention Is All you Need* című tanulmányban 2017-ben. Az ekkor publikált új megközelítés azonnal felhívta magára a figyelmet, hiszen a különböző fordítási teszteken csúcspontszámokat ért el annak ellenére, hogy a modell képzési időigénye és költsége töredéke volt a korábbi

[32] Chernyavskiy–Ilvovsky–Nakov 2021: 677–693.



5. ábra: A transzformátorarchitektúra ábrázolása
Forrás: a szerző szerkesztése VASWANI *et al. 2017: 3 alapján*

típusoknál.[33] A Stanford Egyetem kutatócsoportja a 2021-es publikációban már alapmodellnek nevezte a transzformátorokat, mert álláspontjuk szerint paradigmaváltást hajtanak végre az MI-közegben, ezzel új korszakot megindítva.[34]

A „transzformátor" elnevezés abból eredeztethető, hogy a bemeneti szekvenciát kimeneti szekvenciává alakítják át.[35] Ennek érdekében ezek az architektúrák több (N-számú) rétegből (transzformátorblokkból) állnak, amelyeknek vannak önfigyelő *(self-attention)*, előrecsatolt *(feed-forward)* és normalizáló *(normalization)* rétegei. Ezek a rétegek együttesen dolgoznak annak érdekében, hogy megértsék a bemenetet, és előre megjósolják a várható kimenetet. A transzformátortechnológia a hatékonyságát két innovatív megoldásnak köszönheti, amelyek a *pozicionális kódolás* és az *önmegfigyelés*. Előbbi egy korszerű feldolgozási megoldás (a modell a szekvencián belüli előfordulást használja bemenetként) a szekvenciális (szavak sorrendjében történő) betáplálás helyett. Az önfigyelési mechanizmus a bemeneti adatok feldolgozásakor azokat a bemenet többi részével összefüggésben fontosságuk alapján súlyozza, ennek köszönhetően a modell ezekre összpontosít, és nem kell a teljes bemenetre maximális figyelmet fordítania. A két módszer együttes használata lehetővé teszi az összefüggések hatékony analizálását.[36] A transzformátorok egyaránt tartalmaznak kódoló és dekódoló komponenst is (5. ábra), amelyeket az adott feladat magasabb szintű végrehajtásának érdekében együttesen, de akár külön-külön is képesek

33 VASWANI et al. 2017.
34 BOMMASANI et al. 2021: 3.
35 FERRER 2024.
36 Large Language Models 2024.

használni. A transzformátorhálózatok funkcióit és jellegzetességeit tekintve az alábbiak szerint definiálhatók.

A transzformátormodellek olyan neurális hálózatot tartalmazó nagy nyelvi modell architektúrák, amelyek figyelemmechanizmusokat alkalmaznak a szekvenciális adatok feldolgozására és a kontextusbeli jelentésük megértésére, ezzel hatékonyan képesek nagy mennyiségű szöveget elemezni és generálni.

Az architektúra megjelenését követően számos NLP-feladatban (például szöveg-összefoglalás, kérdés-válaszolás, szöveggenerálás) kiemelkedő teljesítményt ért el, és azóta kulcsszerepet tölt be a nagy nyelvi modellek sikerében.

A tokenizáció és a beágyazás

A transzformátorarchitektúrában alkalmazott neurális hálózatokban a neuronok száma véges, tehát a mélyhálón keresztülhaladó adatfolyamot csak egy fix számú neuron tudja kezelni. Ennek okán a szöveges tartalmat a karakterenként való feldolgozás helyett célszerűbb hatékonyabban feldolgozható formába alakítani. A *tokenizálás* a bemeneti és kimeneti szöveges tartalmak kevesebb számú egységekre történő felosztását jelenti oly módon, hogy a számítási erőforrás és memóriaszükséglet csökkentése mellett a szöveg struktúrája, tartalma és jelentése megmaradjon. A *tokenek* lehetnek szavak, szótöredékek, számok, önálló karakterek, szimbólumok vagy írásjelek.[37]

Többféle tokenizációs eljárás létezik. Az LLaMA[38] tokenizerje a mondatok darabjait dolgozza fel, míg az OpenAI GPT-4[39] modelljének *Byte-Pair Encoding* (BPE) módszere a leggyakrabban előforduló karakter- vagy bytepárokat egyetlen tokenbe vonja össze. Általánosságban alkalmazható az a megállapítás, hogy az angol nyelv esetében egy token körülbelül 4 karakternyi szövegnek felel meg,[40] míg a magyar nyelv esetében ez megfigyelésem alapján 2-3 karakter közé tehető (6. ábra).

Ennek köszönhető, hogy az angol nyelvet előnyben részesítő tokenizátorok estében a bemenetként és kimenetként meghatározott maximum karakterszám magyar szöveg esetében jóval kedvezőtlenebb. A generatív képességek alkalmazása során a szöveges kimenet szintén tokenizált formában realizálódik, így alapesetben a nagy nyelvi modellek által létrehozott szintetikus szöveges tartalmak utólag beazonosíthatók[41] és elkülöníthetők a természetes módon alkotott szöveges tartalmaktól. A tokenizálás során a szövegrészletekhez egyedi azonosítók *(tokenID)* tartoznak (7–8. ábra), amelyek a beágyazás során szavakként egy számsor (vektor) formát vesznek fel (8. ábra).

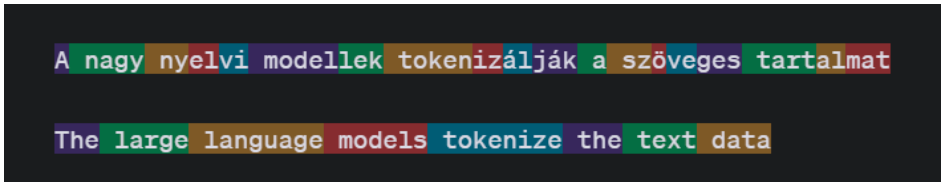
37 Bottyán 2024: 45.

38 Az LLaMA *(Large Language Model Meta AI)* a Meta (korábban: Facebook) által fejlesztett nagy nyelvi modellcsalád neve, amelyet elsősorban kutatási célokra hoztak létre.

39 Az OpenAI GPT-4 az OpenAI által fejlesztett Generative Pre-trained Transformer modellcsalád negyedik iterációja.

40 Humor 2023.

41 A tokenizálásnak köszönhetően a szöveges tartalom analizálható aszerint, hogy MI által lett-e generálva. Ilyen eszköz például az AI Content Detector. Forrás: www.scribbr.com/ai-detector



6. ábra: A GPT-3.5 és a GPT-4 tokenizálásának szemléltetése magyar és angol nyelvű szöveg esetén
Forrás: a szerző szerkesztése OpenAI Platform Tokenizer alapján

```
[32, 73270, 4423, 296, 4627, 2359, 22225, 6602, 482, 6568, 85234, 261,
8515, 573, 737, 4431, 31502, 280, 8872, 279, 976, 4410, 6439, 7015,
192720, 290, 2201, 1238]
```

7. ábra: A 6. ábrán feltüntetett szöveges tartalom egyedi azonosítói (TokenID) a GPT-3.5 és a GPT-4 tokenizátora alapján
Forrás: a szerző szerkesztése OpenAI Platform Tokenizer alapján

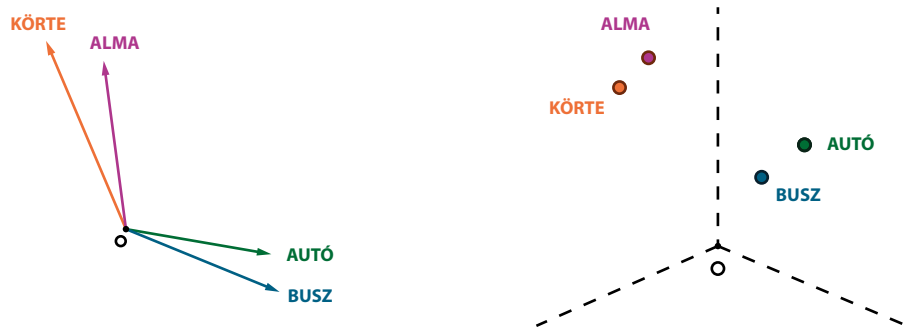
Token szöveg	Token ID	Beágyazott vektorok
'<A>'	32	→ [0.122, -0.346, 0.561, -0.785, …]
'<nagy>'	73270	→ [-0.678, 0.451, 0.895, -0.125, …]
'<ny>'	4423	→ [0.233, -0.898, 0.125, 0.676, …]
'<el>'	296	→ [0.562, 0.341, -0.234, -0.455, …]
'<vi>'	4627	→ [-0.781, 0.675, -0.453, 0.238, …]
'<model>'	2359	→ [0.342, -0.235, 0.894, -0.562, …]
'<lek>'	22225	→ [-0.455, 0.789, 0.129, -0.893, …]

8. ábra: A tokenek azonosítói és beágyazott vektorértékei
Forrás: a szerző szerkesztése

A beágyazás *(embedding)* nem LLM sajátosság, hanem a gépi tanulásban az adatábrázolás egy olyan speciális formátuma, amely megfelel a neurális hálóval történő feldolgozáshoz. Ehhez olyan előre definiált készlet (szótár) szükséges, amely a modellel megismertetni kívánt tokenek (szavak, szótöredékek) listáját tartalmazza.

Az LLM-ek a képzési folyamatban a szavak közötti kontextuális jelentést és kapcsolatot azonosítják, és azt információsűrű lebegőpontos számábrázolás formában a beágyazás folyamataként reprezentálják. A könnyebb érthetőség érdekében (a szóbeágyazások példájánál maradva): a reprezentált szó egy jellemzővektor alkalmazásával olyan numerikus értéket vesz fel, amely számos annyi jellemzőt tartalmaz, amennyit előzetesen megszabtunk. Ezeket a numerikus értékeket (lebegőpontos vektorokat) elhelyezhetjük egy többdimenziós vektortérben. A dimenziók száma a jellemzővektor komponenseivel (n) azonos, tehát ahány szempont szerint vizsgáljuk az adott szót, annyi eltérő dimenzióban szerepeltetett állapotvektort (n számú) kap. Általánosságban elmondhatjuk, hogy a közel azonos jelentésű szavak azonos térben hasonló vektorértékeket vesznek fel, tehát a két vektor közötti távolság a „rokonságukat" méri (9. ábra).

A numerikus reprezentációk (állapotvektorok) vektoradatbázist alkotnak, amelyekben egyszerűen alkalmazhatók a szükséges vektormatematikai (például legközelebbi



9. ábra: A hasonló jelentéssel rendelkező szavak egymáshoz közeli vektorértéket vesznek fel
Forrás: a szerző szerkesztése

szomszéd analízis, koszinuszhasonlóság stb.), valamint a keresési, lekérdezési, osztályozási műveletek és az egyéb nyelvi feldolgozási feladatok (például fordítási és nyelvi transzformációs műveletek) is.

Az utasítástervezés

A számítástechnikában a *prompt* kifejezést a parancssorból vezérelhető operációs rendszerek megjelenése óta alkalmazzák, és gyakorta találkozhatunk vele a nagy nyelvi modellek kapcsán is, hiszen az a modellnek tett utasítást jelenti. Azonban míg kezdetben a kiadott utasításnak szigorú szabályrendszer szerint kellett felépülnie, az LLM-ek esetében már a természetes nyelvi formula használatával rögzített gondolat is utasítás, azonban annak különböző minőségi és mennyiségi alkalmazása természetesen eltérő kimeneteleket eredményez. Az LLM-technológiában rejlő lehetőségek maximális kihasználására alakult ki *az utasítástervezés (prompt engineering), amely a generatív mesterséges intelligencia számára kiadott utasítások megalkotásának hatékony módszertanát jelenti.* Az utasítástervezés tudásanyagának alkalmazása csökkenti a nagy nyelvi modellek leghatékonyabb alkalmazási módja és a felhasználók közti szakadékot.[42]

A transzformátorarchitektúra működési elve alapján a hatékony utasításkiadás követendő fundamentális elvei az alábbiak:

1. *Egyértelműség*: ajánlatos világosan kiadni az utasítást és kerülni a bonyolult, szakzsargont használó megfogalmazásokat, hiszen a legtöbb modell alaptanítása nem tartalmaz tartományspecifikus adatkészletet.
2. *Konkrétság*: a rövid karakterszámú szövegszerű bevitel helyett inkább „hosszú ívű" gondolatokat, tartalmában konkretizált és példázatokat is tartalmazó szövegezés ajánlott, amelyben teljeskörűen, kontextusban testet öltve jelenjen meg a kívánt szükséglet. Ez hozzájárul a figyelemmechanizmus eredményességéhez, ezáltal a pontos kimenet elérésére.[43]

[42] Bottyán 2024: 44.
[43] Bottyán 2024: 44.

Valamint a megfelelő utasításnak kötelező elemei is vannak:

1. *Szerep:* amennyiben követendő szerepkört helyezünk el az utasításban, az támogatja a mechanizmusokat a releváns, eredményes válaszadásban. (Például: „te egy katonai műszaki ismeretekkel is rendelkező építész végzettségű munkatárs vagy").
2. *Utasításleírás:* a konkrét elvárt feladat-végrehajtás leírása, amely világosan meghatározza a kívánt kimenetet. (Például: „listázd nekem egy veszélyes anyag tárolására szolgáló katonai célú létesítmény építészeti felújítása során szóba jöhető kockázatokat. A válaszod legyen listaszerű, minden lista eleme mellé rövid magyarázatot adj, amely legalább egy bekezdésnyi hosszúságú legyen.")
3. *Kérdés:* az információadásra való felkérés. A promptba fogalmazott kérdés kellő figyelmet kap a reprezentálás során, így szignifikánsan formálja a kimenetet. (Például: „milyen kockázatokkal kell számolnom?")
4. *Kontextus:* az adott probléma vagy feladat témakörét átfogóan és összefüggően tárgyaló szöveges tartalom (úgynevezett: „gondolatok láncolata") javítja a kimenet pontosságát. (Például: „Jelenleg egy katonai létesítmény építészeti és műszaki felújításának előkészülete zajlik. Az épület 1975-ben épült, építőeleme tégla és beton. Kétszintes és az alapterülete 2000 m^2, a tető 2018-ban már felújításra került…")
5. *Példázat:* tovább növelhetjük a kimenet sikerességét, ha egyértelműen meghatározzuk a kívánt válasz struktúráját. (Például: „[kockázat 1]: Leírás…; [kockázat 2]: Leírás…"). A példázat alkalmazása a promptban, és azáltal a kimenettel szembeni strukturális követelmények előzetes meghatározása kiemelten fontos, ha az LLM által generált adatot további rendszerekben kívánjuk feldolgozni.[44]

Továbbá a kötelező elemek mellett a hatékonysághoz hozzájárulhat a *COSTARL (context, objective, style, tone, audience, response, length)* módszer alkalmazása is, amely során további fontos információkat szolgáltatunk a modellnek annak érdekében, hogy a generált kimenet nagy valószínűséggel a várt eredménnyel járjon:

- *Összefüggés (context)*: adjunk releváns háttér-információkat, például azt, hogy milyen feladatra vesszük igénybe.
- *Cél (objective)*: egyértelmű utasítások adása arról, hogy milyen tevékenységet végezzen.
- *Stílus (style):* adjunk információt arról, milyen stílusban várjuk a generatív szöveget (akadémiai, vicces, köznyelvi stb.).
- *Hangnem (tone):* ez az utasításrész meghatározza, hogy milyen hangnemet várunk el eredményként (melankolikus, ironikus, humoros stb.).
- *Célközönség (audience):* a célközönség ismerete segít a kontextuskapcsolatok pontos azonosításában, majd ennek megfelelően alakítja a választ.
- *Válasz (response):* ez a rész meghatározza a válasz formátumát (részletes jelentés, táblázat, felsorolás stb.).
- *Hossz (length):* a generált tartalom hosszúságára vonatkozó elvárásunkat feltüntethetjük a promptban (karakterszám, bekezdések száma stb.).

[44] Savio 2024.

A tevékenység angol megfogalmazásával *(prompt engineering)* ellentétben nem nevezhető technikai vagy mérnöki tevékenységnek, de ismeretanyagának alkalmazása elengedhetetlen része az LLM hatékony használatának. Adott esetekben a nagy nyelvi modellek finomhangolása helyett az úgynevezett „prompt programozásra" helyezik a hangsúlyt, amely során a megfelelő utasítás kiválasztása mellett a bemeneti utasítást számottevően módosítják annak érdekében, hogy a kívánt kimenetet érjék el. Eltérő architektúrájú nagy nyelvi modellek némileg eltérően reagálnak azonos promptokra, ezáltal azok specifikálhatók egyes nagy nyelvi modellekre is, manuális vagy automatikus módszerekkel egyaránt.[45]

Paraméterek

A nagy nyelvi modellek kifejezetten komplikált architektúrájú rendszerek, összetettségükre vonatkozó mértékegység a *paraméterszám*. A paraméterek nem a mélyhálózat neuronjai, hanem az azok között létrejövő kapcsolat súlyai, amelyek a neuronok között létrejövő hatásokat határozzák meg. Ezek a számértékek a képzés során, a tanítóadat mintája alapján úgy állítódnak be (kalibrálódnak), hogy megtalálják a tanítóadatban fellelhető kapcsolatokat (például a nyelv szerkezete, szintaxisa, szemantika stb.). A nagy nyelvi modellek paramétereinek számos típusa (például beágyazási, figyelem, kimeneti, pozíciókódolási, normalizáló stb.) együttműködik annak érdekében, hogy a modell emberszerű szöveget generáljon. A teljesítmény optimalizálása során beállításuk kritikus fontosságú, ezért jelentős kísérletezés előzi meg a megfelelő kalibrációs állapot elérését. A transzformátor-architektúrákról általánosságban elmondható, hogy minél nagyobb a paraméterszám, annál összetettebb modellről beszélhetünk, valamint annál hatékonyabb mintafelismerésekben és precízebb a kimenet-előállításban, ám a paraméterszám növelése nem minden esetben vezet automatikusan jobb teljesítményhez.

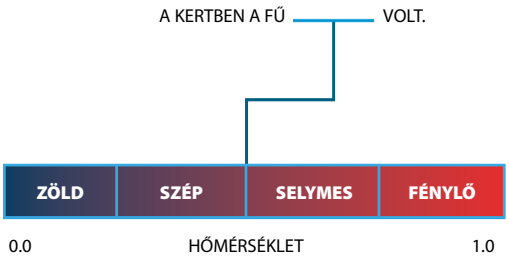
A modellek paraméterszámának jelölése gyakorta B jelzéssel (például Llama 2 13B, Llama 2 70B) történik, amely az adott modell paraméterszámának milliárd *(billion)* mennyiségét jelöli. Ezt a jelölést törvényszerűen alkalmazzák akkor, ha az adott modellből több, eltérő paramétszámmal rendelkező verzió elérhető. Természetesen minél több paraméterrel rendelkezik egy modell, annál nagyobb a tárhelyigénye (1. táblázat), valamint képzése és üzemeltetése annál erőforrás-igényesebb.

1. táblázat: A Llama nagy nyelvi modell néhány típusának paraméterszáma és tárhelyigénye

Modell	Paraméter	Tárhelyméret
Orca Mini 3B	3 milliárd	1,9 GB
Llama 2 7B	7 milliárd	3,8 GB
Llama 2 13B	13 milliárd	7,3 GB
Llama 2 70B	70 milliárd	39 GB

Forrás: a szerző szerkesztése

[45] Shin et al. 2020: 1–2.



10. ábra: A hőmérséklet értéke befolyásolja a kapott szöveg változatosságát
Forrás: a szerző szerkesztése

Bizonyos paraméterek *(hiperparaméterek)* értékei a tanítási folyamatot követően, utólag is kalibrálhatók. A szolgáltatásként elérhető modellek esetében a kalibrációs beállítások általában a szolgáltatást igénylő számára is elérhetőek, tehát adott feladatra szabható a modell működése, így jobb minőségű szolgáltatás elérhető. Ilyen hiperparaméterek az alábbiak.

A hőmérséklet *(temperature)* szabályozza, hogy a generatív folyamat során mekkora szerepe legyen a véletlenszerűségnek. Alacsony hőmérséklet esetén általában unalmas és kiszámítható, ugyanakkor koherens és konzisztens szöveget kapunk. Közepes értékként változatosabb, de még észszerű és logikus szöveg az eredmény. Magas értéknél nagyon ötletes és kalandos, de kiszámíthatatlan szöveges kimenetet is kaphatunk.

A *Top-P* paraméter szintén a szöveg változatosságára van hatással, kiválasztja, hány token legyen kijelölve a következő helyre további számítások céljából. Az alacsony Top-P nagyobb pontosságot és megbízhatóságot, míg a magasabb érték kreatívabb kimenetet eredményez.

A jelenléti büntetés *(presence penalty)* paraméter a generált szövegben szabályozza a szavak és kifejezések megjelenését. Magas értéket alkalmazva új témák, kifejezések, szinonimák jelennek meg, míg alacsonyabbal több szóismétlés és sablonosság várható.

A gyakoriság büntetés *(frequency penalty)* paraméter hasonlóan működik a jelenléti büntetéshez, azonban figyelembe veszi a promptban található tokeneket is. Amelyik token gyakorisága jelentős, azt bünteti a paraméter, így csökkentve a megjelenési valószínűséget. Magas gyakoriságbüntetés-kalibrációval egyedibb és változatosabb szöveget lehet elérni, míg alacsony beállítással akár értelmetlen szöveget is kaphatunk.[46] A paraméterek beállítására nem találunk követendő szabályokat, általában egymással létrejövő harmóniára és az egyensúlyra kell törekedni kalibrációjuk során, amelyet többnyire nagyszámú kísérletezéssel érnek el.

Összegzés

A tanulmány első részében a mesterséges intelligencia fogalmára alapozva, a gépi tanuláson és a természetesnyelv-feldolgozáson át mutattuk be a nagy nyelvi modellek legáltalánosabb

[46] Ehab 2023.

architektúravariánsait, valamint a teljesség igénye nélkül a modellek működése mögött található alapvető technikai megvalósításokat. Emellett a nagy nyelvi modellek és a transzformátorarchitektúrával kapcsolatos alapvető gondolatok magyar nyelvű meghatározásban öltöttek alakot. A modellek utasításvezérlésének módszertana, valamint annak kalibrációs lehetőségei szintén fontos fundamentumai a nagy nyelvi modellek leghatékonyabb felhasználási lehetőségeinek. Az e tanulmányban összeállított átfogó ismeretanyagban bízva az remélhetőleg jó szolgálatot tesz a jövőben azon kutatóknak, akik általános céllal, vagy kifejezetten a nemzetbiztonságot érintő információfeldolgozási eljárások fejlesztésének céljából nyúlnak hozzá ezen innovatív technológia alapjaihoz. Bizonyos, hogy a LLM-technológia megszerezte létjogosultságát az információfeldolgozásban, és bár a közvetlen és eredményes felhasználása egyelőre még körülményes, tőkeigényes és jelentős tudományos erőforrásokat is igényel, átlátható folyamat. A tanulmány következő részében erre, vagyis a modellek képzési folyamatára helyeződik majd a hangsúly, továbbá az azzal kapcsolatos szükségletek bemutatására. A tanulmánykettős záró részében összefoglalásként a nagy nyelvi modellek alkalmazhatósági területeinek említése mellett egy SWOT-elemzés keretében a releváns belső és külső tényezők kiértékelése fog megtörténni annak érdekében, hogy egy szervezeti alkalmazás esetén a modell felhasználása a leghatékonyabban történjen meg a fellépő kockázatok kezelése mellett.

Felhasznált irodalom

Agüera y Arcas, Blaise (2022): Do Large Language Models Understand Us. *Daedalus,* 151(2), 183–197. Online: https://doi.org/10.1162/daed_a_01909

Az Európai Parlament és a Tanács (EU) 2024/1689 rendelete (2024. június 13.) a mesterséges intelligenciára vonatkozó harmonizált szabályok megállapításáról, valamint a 300/2008/EK, a 167/2013/EU, a 168/2013/EU, az (EU) 2018/858, az (EU) 2018/1139 és az (EU) 2019/2144 rendelet, továbbá a 2014/90/EU, az (EU) 2016/797 és az (EU) 2020/1828 irányelv módosításáról (a mesterséges intelligenciáról szóló rendelet).

Bommasani, Rishi et al. (2021): *On the Opportunities and Risks of Foundation Models. Center for Research on Foundation Models.* Stanford Institute for Human-Centered Artificial Intelligence, Stanford University. Online: https://arxiv.org/pdf/2108.07258.pdf

Bottyán Sándor (2024): *Nagy nyelvi modellel támogatott nyílt forrású információgyűjtés a kibertérben.* Tudományos diákköri dolgozat. Budapest: Nemzeti Közszolgálati Egyetem Államtudományi és Nemzetközi Tanulmányok Kar.

Chernyavskiy, Anton – Ilvovsky, Dmitry – Nakov, Preslav (2021): Transformers: „The End of History" for Natural Language Processing? In *Machine Learning and Knowledge Discovery in Databases.* Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17,Z021, Proceedings, Part III 21. Springer, 677–693. Online: https://doi.org/10.1007/978-3-030-86523-8_41

Chowdhary, K. R. (2020): *Fundamentals of Artificial Intelligence.* Springer India. 1st ed. Online: https://doi.org/10.1007/978-81-322-3972-7

Devlin, Jacob et al. (2018): *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* Online: https://doi.org/10.48550/arXiv.1810.04805

Ehab, Michael (2023): The Secrets of Large Language Models Parameters: How They Affect the Quality, Diversity, and Creativity of Generated Texts [With Examples]. *Medium,* 2023. július 30. Online: https://michaelehab.medium.com/the-secrets-of-large-language-models-parameters-how-they-affect-the-quality-diversity-and-32eb8643e631

Ferrer, Josep (2024): How Transformers Work: A Detailed Exploration of Transformer Architecture. *Datacamp,* 2024. január 9. Online: www.datacamp.com/tutorial/how-transformers-work

Humor, Michael (2023): *Understanding „Tokens" and Tokenization in Large Language Models*. Online: https://blog.devgenius.io/understanding-tokens-and-tokenization-in-large-language-models-1058cd24b944

Malik, Farhad (2019): Neural Networks – A Solid Practical Guide. Explaining How Neural Networks Work With Practical Examples. *Medium,* 2019. május 16. Online: https://medium.com/fintechexplained/neural-networks-a-solid-practical-guide-9f343594b02a

Munoz, Andres (2012): *Machine Learning and Optimization*. Online: www.semanticscholar.org/paper/Machine-Learning-and-Optimization-Munoz/7fbba79630b5a09dd66ab13f00c3aefaa56cf268

Ongsulee, Pariwat (2017): *Artificial Intelligence, Machine Learning and Deep Learning*. 15th International Conference on ICT and Knowledge Engineering (ICT&KE). Bangkok. Online: https://doi.org/10.1109/ICTKE.2017.8259629

Raffel, Colin et al. (2020): Exploring the Limits of Transfer Learning With a Unified Text-To-Text Transformer. *The Journal of Machine Learning Research,* 21(1), 1–67. Online: https://jmlr.org/papers/volume21/20-074/20-074.pdf

Savio, Jacob (2024): What Is Prompt Engineering? Definition, Elements, Techniques, Applications, and Benefits. *Spiceworks,* 2024. április 24. Online: www.spiceworks.com/tech/artificial-intelligence/articles/what-is-prompt-engineering

Shin, Taylor et al. (2020): AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In Webber, Bonnie et al. (szerk): *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 4222–4235. Online: https://doi.org/10.18653/v1/2020.emnlp-main.346

Vaswani, Ashish et al. (2017): *Attention Is All You Need*. Online: https://doi.org/10.48550/arXiv.1706.03762