

Kezdő programozásoktatás támogatása mesterséges intelligenciával

Bende Imre

beiraai@inf.elte.hu

ELTE Eötvös Loránd Tudományegyetem Informatikai Kar

Absztrakt. A mesterséges intelligencia térnyerésével és gyors, folyamatos fejlődésével elkérülhetetlen, hogy az az oktatás és azon belül is a programozásoktatás részévé is váljon. Jelen kutatás során annak lehetőségeit mutatom be, hogy a mesterséges intelligencia alapú eszközök miként alkalmazhatóak a kezdő, programozási tételt igénylő feladatok megoldásának támogatására. Kutatásomban azt vizsgálom, hogy a jelenleg könnyen elérhető eszközök képesek-e felismerni a megoldáshoz szükséges programozási tételeket, meg tudják-e állapítani egy megoldásról, hogy az helyes-e, valamint képesek-e ezen feladatok megoldására adott programozási nyelven.

Kulcsszavak: programozásoktatás, mesterséges intelligencia, ChatGPT, Bard

1. Bevezetés

A ChatGPT 2022-es megjelenése a köztudatba hozta a mindenki számára ingyenesen elérhető mesterséges intelligencia (rövidítve: MI) alapú eszközök használatát. Az oktatás területén a témában elsősorban irodalmi áttekintések és kommentár műfajú cikkek jelentek meg, ezek mellett ugyanakkor kiőbb számban kvalitatív, a ChatGPT-vel végzett kérdés-válasz jellegű kutatások is születtek [8]. Egyelőre sokakban szkepticizmus és kezdeti félelem társul a mesterséges intelligenciához [7], viszont elkérülhetetlen, hogy az idő előrehaladtával, annak folyamatos fejlődésével a mindennapjaink részévé váljon, így kiemelten fontos, hogy az oktatás során azt felhasználjuk, sőt, meg is értsük a fiatalokkal annak működését.

Az eddigi általános csoportszintű oktatási formák a mesterséges intelligencia alapú eszközökkel, azon belül is a ChatGPT segítségével személyesebbé tehetőek, valamint az MI a tanárok terhelését is csökkentheti azáltal, hogy adott kérdésekre azonnali választ tud adni [1,6,11]. Azonban míg a mesterséges intelligencia által támogatott oktatás és tanuláson alapuló rendszerek (AIED) jobban teljesítenek nagyobb osztályokban, mint a tanárok, addig egyéni oktatás során kisebb mértékben rosszabb eredmény születt [2].

A programozásoktatás során a ChatGPT-vel különböző feladatokra használható. Ilyenek lehetnek a következők:

- Fogalmak, algoritmusok megértése
- Kód generálása
- Hibaellenőrzés, hibakeresés (hibajavítás)
- Kód optimalizálás [9]

Több kutatás utal arra, hogy a mesterséges intelligencia oktatásba történő integrálásának pozitív hatásai vannak: egy kontroll csoportos vizsgálat esetén a ChatGPT-t használóknál szignifikánsan javult az algoritmikus gondolkodásuk, programozási készségeik, valamint a motivációjuk [12], míg egy esettanulmány pedig arról számolt be, hogy az eszköz használatát követően a résztvevők többségénél javultak a kognitív képességeik oly módon, hogy csökkent a kódoláshoz szükséges idő, így több idejük maradt az algoritmus átgondolására, problémamegoldásra [13]. Emellett a ChatGPT használata segíti a diákokot abban, hogy kódjuk olvashatóbb legyen (alkalmazása közben több komment jelent meg a

forráskódban, illetve a változók névadásában és a behúzások használatában is pozitív eredmény lett) [10].

Jelen tanulmányban azt vizsgálom, hogy milyen módon támogatható a kezdő programozásoktatás mesterséges intelligencia alapú eszközök segítségével. Kutatásom során a következő kutatási kérdéseket vizsgáltam az OpenAI ChatGPT és a Google Bard felhasználásával:

- Képes-e adott feladatszöveg alapján meghatározni a megoldáshoz szükséges programozási tételt?
- Képes-e a feladatszöveg és az implementáció alapján meghatározni annak helyességét?
- Képes-e a feladatszöveg és hibás implementáció alapján a hibát felismerni és annak javítására tanácsot adni?
- Képes-e adott feladatszöveg alapján implementálni annak megoldását?
- Képes-e személyre szabott feladatokat kitalálni, amelyek megoldásához adott programozási tételre van szükség?

Mindkét említett eszköz folyamatos fejlődésen megy keresztül, viszont vannak hiányosságaik, hibáik [4], ami miatt a kritikus gondolkodás mindenképpen fontos a válaszok értelmezése közben.

2. Felhasznált eszközök

Jelen tanulmányban két eszközt használtam: az OpenAI által fejlesztett ChatGPT-t és a Google Bard nevű chatbotját. Mindkettő könnyen elérhető mindenki számára, habár regisztrációhoz kötött a használatuk, azonban ezt követően ingyenesek, valamint magyarul is használhatóak. Azért is esett ezekre a választásom (azon felül, hogy jelenleg ezek a legnépszerűbb, leggyakrabban emlegetett eszközök a témában), mert a magyar oktatásban külön idegennyelv-ismeret nélkül használhatóak, valamint csupán egy böngészőre és internetkapcsolatra van szükség ezek elindításához, így könnyen integrálhatóak lehetnek az oktatásba. A tanulmány írásakor a ChatGPT 3.5-ös verziója az, ami regisztrációt követően ingyenesen használható, viszont fizetős szolgáltatás részeként a 4.0-ás verzió is elérhető már.

Habár a kísérlet során nem használom, mindenképpen fontosnak tartom megemlíteni a GitHub Copilot nevű fejlesztői környezetbe építhető eszközt is, mely kérdés/utasítás alapján generál kódot. Erős támogatottsága miatt az eszköz népszerűsége folyamatosan növekszik, a szoftverfejlesztés szereplőinél is egyre több ízben jelenik meg a felhasználhatóságának vizsgálata, témaköre [14], mely a jövőben csökkentheti a szoftverfejlesztés idejét. Az eszköz nem képezi a vizsgálat részét, ugyanis jelenleg magyar nyelvű kérdésekre nem tud válaszolni, illetve előfizetés szükséges a használatához (oktatók, diákok ingyenesen használhatják, viszont a regisztrációs folyamatot emiatt meghosszabbítja egy jóváhagyási időintervallum is).

Emellett egyre többen próbálják kihasználni a GPT lehetőségeit, ilyenre példa a GPTutor, mely Visual Studio Code bővítményként képes részletesebb, kevesebb hibával rendelkező kódmagyarázatokat adni a megadott implementációkra [3].

A kutatás során az eszközök programozási tételekkel¹ kapcsolatos ismereteit vizsgálom, így a tanulmányban egy-egy tétel alatt az alábbi feladatokra adott válaszokat, megoldásokat értem:

- Egy papírgyűjtési akcióban mindenkiről feljegyezték, hogy ki hány kiló papírt hozott. Írj programot, ami megadja, hogy összesen mennyi papírt gyűjtöttek a résztvevők!

¹ Jelen tanulmányban az egy sorozaton értelmezett programozási tételeket vizsgáltam, melyek az elemi programozási tételek: összegzés, megszámlálás, maximum-kiválasztás, eldöntés, kiválasztás, keresés, valamint az összetett programozási tételek közül a másolás, kiválogatás és a szétválogatás.

- Egy N (konstans érték) fős baráti társaság egy nyaralás minden napján feljegyezte, hogy aznap mennyit költöttek. Készíts programot, ami meghatározza, hogy hány olyan nap volt, amikor fejenként 100 eurónál is többet költöttek.
- Egy fesztivál minden napján feljegyezték, hogy hány belépőt adtak el aznap. Készíts programot, ami meghatározza, hogy melyik napon voltak a legtöbben a fesztiválon!
- A vidámparkban minden játékért fizetni kell. Készíts programot, ami a játékok ára alapján, eldönti, hogy van-e olyan játék, ami kevesebb, mint 500Ft-ért igénybe vehető!
- Adott egy számokat tartalmazó $A(N)$ tömb és egy B szám. Adjunk meg két olyan elemet az A -ból, amelyek szorzata éppen a B !
- Adott egy $A(N)$ vektor. Elemei közül a negatívok helyére írjunk 0-t!
- Adottak egy futóverseny eredményei, valamint az első osztályú szinthatár. Kik értek el első osztályú eredményt?
- A BKV utasszámlálást végzett. Ismerjük, melyik járaton hány férőhely van és átlagosan hányan utaztak rajta. Zsúfolt az a busz, ahol a férőhelyek legalább 80%-a foglalt. Üres, ha 20%-nál kisebb a kihasználtság. Készítsünk programot, mely a járatokat szétválogatja zsúfolt, üres és egyéb buszok szerint!

A céloom rövid, egyszerű feladatok kiválasztása volt, amelyek könnyen értelmezhetőek, megoldhatóak és a kezdők számára is jó bevezetés lehet a programozási tételek világába.

Minden szempont vizsgálatánál egy táblázatban jelölöm, hogy az adott eszköz az adott problémát sikeresen teljesíti-e. Pipa jelzi azt, hogy helyes választ adott az eszköz, a mínusz a helytelen választ mutatja, míg a kérdőjel azt jelzi, hogy ha részben helyes a válasz, de nem teljes mértékben.

3. Röviden a prompt tervezésről

A tanulmánynak nem célja a prompt tervezés részletes ismertetése, bemutatása (részletesebben ld. [5]), viszont vannak tulajdonságok, amelyek ismerete szükséges ahhoz, hogy átlássuk milyen kérdéseket szükséges feltenni az általunk választott eszközöknek, illetve miként tudjuk módosítani azokat, hogy a válasz minél pontosabb és a későbbiekben felhasználható lehessen.

Fontos, hogy az utasítás, kérdés pontosan, részletesen legyen megfogalmazva, ezáltal ténylegesen arra kapjunk választ amire szeretnénk. Ellenkező esetben hosszú, irreleváns információkat is kaphatunk a válaszban. Jelen esetben többnyire rövid kérdéseket írok, de módosítható, pontosítható, hogy az eszköz mely programozási nyelvben adja meg a megoldást, mennyire legyen részletes a válasz, tartalmazzon-e mellékes információkat, illetve mennyire ellenőrizze a bemenő adatokat.

Bizonyos értelemben mindkét eszköz emlékszik a kontextus ablak tartalmára, így ezzel lehetőségünk van egy adott kérdéssorhoz tartozó alap információk megadására. A programozási tételek felsorolásával, ismertetésével pont ezt használjuk ki, így ezáltal a későbbi kérdések megválaszolásánál pontosak lesznek azok megnevezései.

A válaszok helyességénél mindig kritikusnak kell lenni. Az MI eszközök nem adnak minden esetben helyes választ, sőt bizonyos esetekben olyan információkat hoznak létre, amelyek nem is léteznek (hallucináció). Részben pont a helyesség mérése a kutatás célja a programozási tétel igénylő feladatok esetén, így erre nagy hangsúlyt kell fektetni minden esetben.

4. Programozási tételek felismerése a feladatokban

4.1. Programozási tételek felismerése a feladatokban

Első lépésben azt a kérdést tettem fel, hogy mely programozási tételeket ismeri az adott eszköz. A Bard felsorolta az összegzést, maximum-kiválasztást (maximum elem néven), kiválasztást, keresést, rendezést, halmazműveleteket, valamint több adatszerkezet és egyéb informatikai fogalom is megjelent a listában. A ChatGPT elsősorban rendezéseket sorolt fel, emellett megjelent a maximum keresés, szűrés, halmazműveletek. Ezt követően különböző feladatokon keresztül vizsgáltam meg, hogy felismerik-e az eszközök, hogy mely programozási tételt kell felhasználni a megoldásukhoz (ld. 1. táblázat). A kérdést a következő szerkezetben adtam meg:

Prompt: *Melyik programozási tételre van szükség a következő feladat megoldásához: <<feladat szövege>>*

A Bard esetén több fals eredmény született, több ízben megjelent a „vágányváltás” fogalma, mely egy adott feltétel alapján ágazik el. Emiatt új kontextus ablakban ismertettem vele, mely programozási tételek ismertek, amelyek közül ki kell választania a megfelelőt:

Prompt: *Ismertek a következő programozási tételek: összegzés, megszámlálás, maximum-kiválasztás, eldöntés, kiválasztás, keresés, másolás, kiválogatás, szétválogatás. Társítsd a következő feladatokhoz, hogy mely programozási tételre van a szükség a megoldásukhoz!*

Az általa helyesnek vélt programozási tétel említését követően egy Python-ban írt implementációt is ad, példabemenettel. Az egyszerűbb tételt igénylő feladatoknál sikeresen jelölte meg a megoldásokat, viszont az utolsó háromnál, amikor a kimenet egy tömb, már nem a hagyományosnak tekinthető tételt említi. Megjelennek a ciklus, helyettesítés, összehasonlítás fogalmai, valamint utóbbi két feladatnál a kiválasztás is, melyet a kiválogatás szinonimájaként használ, holott a programozás oktatásában ez egy külön tételt jelöl.

A ChatGPT esetében is hasonlóak voltak a válaszok szerkezetei: megjelölte a használandó programozási tételt, majd pszeudókódszerűen, illetve Python-ban azt ki is fejtette. A ChatGPT esetén is szükség volt a tételek ismertetésére, mivel habár több feladatot meg tudott oldani, előjöttek olyan szinonimák, hasonló jelentésű kifejezések használata, ami igényelte a pontosítást. A tételek felsorolását követően már lényegre törőbbek lettek a válaszok is.

	Bard	ChatGPT
Összegzés	✓	✓
Megszámlálás	✓	✓
Maximum-kiválasztás	✓	✓
Eldöntés	✓	✓
Keresés	✓	✓
Másolás	X	✓
Kiválogatás	?	✓
Szétválogatás	?	✓

1. táblázat: Képes-e a ChatGPT és a Bard adott feladatszöveg alapján meghatározni a megoldáshoz szükséges programozási tételt?

4.2. Helyes megoldások ellenőrzése

A következő vizsgálat azt a célt szolgálta, hogy az MI egyszerűbb programozási tételt igénylő feladatok leírásuk és C#-ban megírt megoldásuk alapján meg tudja-e határozni azok helyességét (ld. 2. táblázat). Ennek ellenőrzésére a következő szerkezetben adtam meg a kérdést:

Prompt: *Adott egy programozási feladat: <<feladat szövege>>. Határozd meg igennel, vagy nemmel, hogy a következő C# kód megoldja-e ezt a feladatot: <<C# implementáció>>*

A Bard minden esetben helyesen megválaszolta, hogy az adott implementáció megoldja-e a feladatot, néha plusz tanácsokat is leírt (akár javított megoldás kódjával ellátva), valamint a kódot részletesen elemezte is. Javításként több ízben említette az ellenőrzött beolvasás lehetőségét, adott esetben a feladatban felmerülő kérdés szöveges megválaszolását is preferálta. Ezek a kiegészítések segítségül szolgálhatnak a diákoknak, hogy megértsék egy-egy adott algoritmus működését, ezáltal a hibajavításban is szerepet játszhat. Néhány esetben azonban felesleges többletinformációkat jelölt meg, ilyen például, hogy megszámlálós feladatoknál lehetett volna maximumot is számolni, vagy másolásnál a módosított értékek számságát, holott ezeket a feladat nem kérte.

ChatGPT: a válasz sok esetben lényegre törőbb volt, csak abban az esetben jelent meg kód benne, amennyiben jelentősebb hiba javítását, pontosítást próbálta bemutatni. Ilyen például, hogy az egyik feladat a nap sorszámát kéri vissza, amelynél jelzi a sorszám ([1;N]) és az index ([0;N-1]) közti különbséget.

Mindezek alapján elmondható, hogy jó megoldások esetén mindkét eszköz képes volt felismerni azok helyességét.

	Bard	ChatGPT
Összegzés	✓	✓
Megszámlálás	✓	✓
Maximum-kiválasztás	✓	✓
Eldöntés	✓	✓
Keresés	✓	✓
Másolás	✓	✓
Kiválogatás	✓	✓
Szétválogatás	✓	✓

2. táblázat: Képes-e a ChatGPT és a Bard feladatszöveg és az implementáció alapján meghatározni annak helyességét?

4.3. Helytelen megoldások ellenőrzése

A helyes megoldások áttekintését követően megvizsgáltam, hogy az eszközök a specifikus hibákat mennyire képesek észlelni (ld. 3. táblázat). A következő módosításokat vizsgáltam meg (zárójelben rövid kifejezéssel jelzem, hogy a táblázatban miként utalok rájuk):

- Összegzésnél „+” helyett „-” (Ö1)
- Összegzésnél 0-tól különböző kezdőérték használata (Ö2)
- Megszámlálásnál feltétel invertálása (MSZ1)

- Maximum-kiválasztásnál relációs jel megfordítása (ebből kifolyólag minimum-kiválasztást csinál az algoritmus) (MK1)
- Indexelési hiba miatt, csak N-1 darab elem megy végig az algoritmus (I1)
- Szétválogatásnál a feltételt nem teljesítő elemek listába tárolásának kihagyása (SZ1)
- Pontosvessző kihagyása, ami fordítási hibát ad (F1)

A programozási tétel algoritmusában történő módosításokat követően többségében mindkét eszköz képes volt a hibák felismerésére, közlésére, illetve az azok kijavításához szükséges lépések ismeretetésére is. A Google Bard az MK1-et helyesnek ítélte, habár mutat egy finomítást, amiben ezt a hibát már kijavította.

	Bard	ChatGPT
Ö1	✓	✓
Ö2	✓	✓
MSZ1	✓	✓
MK1	?	✓
I1	✓	✓
SZ1	X	✓
F1	X	✓

3. táblázat: Képes-e Bard és a ChatGPT a feladatszöveg és hibás implementáció alapján a hibát felismerni és annak javítására tanácsot adni?

4.4. Feladatok megoldása

Ezt követően azt vizsgáltam meg, hogy az MI meg tudja-e oldani a fent ismertetett feladatokat (ld. 4. táblázat). Ebben az esetben a következő szerkezetben adtam meg a kérdést:

Prompt: *Oldd meg a következő feladatot C# programozási nyelven: <<feladat szövege>>. A megoldásban szerepeljen a beolvasás és a feladat kérdésére vonatkozó válasz kiírása is!*

Az első feladatok megoldásánál megjelent az a különbség, hogy a ChatGPT egy ciklusba összevonta a beolvasást és az algoritmust, míg a Bard ezeket elkülönítette. Kezdő programozásoktatás során külön szoktuk ezeket kezelni, hiszen így átláthatóbb marad, hogy melyik rész milyen célt szolgál, bár problémát az egybevonás sem okoz.

A Bard a megszámláláshoz tartozó feladatnál nem értelmezte megfelelően azt, hogy külön konstans érték szerepel benne, így a megoldás sem volt megfelelő. Érdekes, hogy az eldöntésnél mindkettő a while ciklus helyett break-et használ, ami, habár ugyanúgy megoldja a feladatot, viszont az oktatás során inkább annak elkerülését tanácsoljuk (főleg kezdő programozásoktatás esetén). A keresésnél jön elő az első nagy különbség: a Bard, ha talál feltételt teljesítő elempárt, akkor kilép, viszont a ChatGPT az összes találatot kiírja (ami által nem is megfelelő a megoldás).

A szétválogatáshoz tartozó feladat megoldásából kiderült, hogy a nyelvi elemekkel jelölt intervallumjelzéseket nem feltétlenül tudja jól értelmezni egyik eszköz sem: a kisebbnél az intervallum szélső értéke nem szabadna, hogy az eredménybe beletartozzon, viszont a legalább kifejezésnél igen. Mindkét eszköz mind a két előbb említett esetben engedte, hogy az intervallum határán lévő elemek benne legyenek az adott halmazban.

Az egyszerűbb leírással rendelkező feladatokat többnyire mind a két eszköz sikerrel megoldotta, így ezt követően megvizsgáltam, hogy hogyan válaszolnak a részletesebben megfogalmazott feladatleírásokra, amelyek tartalmazzák a feladat szövegét leíróan, részletesen specifikált bemenetet, kimenetet, valamint példát is.

Ebben az esetben romlott a helyes válaszok aránya. Eldöntés feladatnál a ChatGPT egy olyan kódot adott meg, mely fordítási hibával rendelkezett: hiányzott egy szögletes zárójel („`int.Parse(input[2])`”), apró javítás után viszont maximális pontszámot kapott. Keresésnél egyáltalán nem vette figyelembe, hogy amennyiben nincs találat, -1-et adjon vissza. A Bard többször túlkomplikált kódot adott, amelyek nem is működtek megfelelően, több esetben a beolvasás nem volt megfelelő (több helyett egy cikluson keresztül ugyanazon változók értékeit módosítja csak), valamint a kiíratásnál az index és sorszám közti különbséggel is voltak problémái.

Habár mindkét eszköznél magyarul tettem fel a kérdéseimet, a Bard minden esetben próbált magyar beszédes kiírásokat tenni a beolvasások elé, miközben angol változóneveket használt, eközben a ChatGPT olyan programot írt, mely kizárólag a feladatot próbálja megoldani, viszont a változónevek magyar kifejezéseket takartak (ezen persze lehet módosítani pontosabb kérdésfeltevéssel).

	Rövid		Részletes	
	Bard	ChatGPT	Bard	ChatGPT
Összegzés	✓	✓	X	✓
Megszámolás	X	✓	✓	✓
Maximum-kiválasztás	✓	✓	X	✓
Eldöntés	✓	✓	X	✓*
Keresés	✓	?	X	X
Másolás	✓	✓	X	X
Kiválogatás	✓	✓	X	✓
Szétválogatás	✓	✓	-	-

4. táblázat: Képes-e a ChatGPT és a Bard adott feladatszöveg alapján implementálni annak megoldását?

4.5. Személyre szabott feladatok létrehozása

Végül azt vizsgáltam meg, hogy az eszközök képesek-e adott témájú, személyre szabott feladatot kreálni meghatározott programozási tételre építve (ld. 5. táblázat). Ebben az esetben a következő formában adtam meg a kérést:

Prompt: *Adj egy programozási feladatot, amelynek megoldásához a <<programozási tétel>> programozási tételre van szükség. A feladat szóljon a <<választott, személyre szabott témakör>> világában. A feladat szövegéhez társuljon egy minta bemenet és kimenet is. Adj egy mintamegoldást is a feladat megoldására <<választott programozási nyelv>> programozási nyelven!*

Az összetettebb állítás miatt azt fogadtam el megfelelő válasznak, ha az eszköz generált egy az adott témakörbe illő feladatot, melynek megoldásához a meghatározott programozási tételre van szükség, amihez megfelelő tesztadat és implementáció társult.

A ChatGPT egy feladat kivételével megfelelően működött: a keresés esetén eldöntésre adott feladatot, tehát csak azt nézte van-e feltételt teljesítő elem, hogy melyik az azt nem.

A Bard esetén több hibával szembesülünk: Maximum-kiválasztás helyett, megszámlálás, keresés helyett maximum-kiválasztásos, kiválogatás helyett maximum-kiválasztásos, illetve szétválogatásos feladatot adott. Szétválogatásnál a feladat leírása megfelelő volt, viszont a megoldásban több hiba is szerepelt: az eredeti koncepciója az volt, hogy két külön tömbbe teszi az értékeket, az egyikbe az elejétől, másikba hátulról viszont indexelési hibák miatt ez nem jól működött.

Másolásnál egyik sem volt igazán kreatív, mivel olyan feladatot találtak ki, melyek változtatás nélkül másolták át az értékeket egy másik tömbbe, habár a kérést megfelelően teljesítették, hiszen a másolás programozási tételre illeszkednek ezek a feladatok.

	Bard	ChatGPT
Összegzés	✓	✓
Megszámlálás	✓	✓
Maximum-kiválasztás	X	✓
Eldöntés	✓	✓
Keresés	X	X
Másolás	✓	✓
Kiválogatás	X	✓
Szétválogatás	X	✓

5. táblázat: Képes-e a ChatGPT és a Bard programozási tételre épülő feladatot kitalálni?

5. Összegzés

	Bard	ChatGPT
Programozási tétel felismerése	5/8	8/8
Helyes megoldások ellenőrzése	8/8	8/8
Helytelen megoldások ellenőrzése	4/7	7/7
Feladatok megoldása (rövid, részletes)	7/8, 1/7	7/8, 5/7
Feladatok létrehozása	4/7	6/7

6. táblázat: Eredmények összegzése

A kutatás eredményeiből (6. táblázat) látszik, hogy a ChatGPT 3.5-ös verziója használható a programozási tételt igénylő feladatok oktatásában: felismeri, hogy mely tételre van szükség a feladat megoldásához, meg tudja őket oldani, képes egy-egy feladathoz tartozó megoldást ellenőrizni, a benne lévő kisebb hibákat javítani, valamint személyre szabott programozási tételt igénylő feladatokat is ki tud találni. Összetettebb leírással rendelkező feladatoknál, habár pár kivétellel, de sikerrel vette az akadályt. Ezzel kimondható, hogy egyszerűbb programozási feladatoknál használható, hasznos oktatási segéd-eszköz lehet a ChatGPT.

A Google Bard még jelenleg nagyon új, ebből kifolyólag a kutatás során hiányosságok is előjöttek, de valószínűsíthető, hogy a terület gyors fejlődésének köszönhetően ez is javulni fog a jövőben. Ez az eszköz több vizsgált kérdéskör esetén adott helytelen választ, ezen belül leginkább az összetett leírású feladatoknál látszódtak helytelen, túlkomplikált implementációk.

Érdeemesnek tartom a jövőben megvizsgálni más programozási területeken belül milyen módon lehetne MI alapú eszközöket beépíteni a tanulásba, oktatásba (például nehezebb, komplexebb feladatok megoldásánál, vagy webes technológiákkal való ismerkedésnél), majd ezek hatékonyságát is mérni (például kontrollcsoportos vizsgálattal).

Irodalom

1. H. Alexandara: *Role of AI in Education*. Interdisciplinary Journal and Humanity (INJURITY), 2, 260-268 (2023). doi: <https://doi.org/10.58631/injurity.v2i3.52>
2. B. du Boulay: *Artificial intelligence as an effective classroom assistant*. IEEE Intelligent Systems, 31(6), 76–81 (2016). doi: <https://doi.org/10.1109/MIS.2016.93>
3. E. Chen, R. Huang, H.-S. Chen, Y.-H. Tseng, L.-Y. Li: *GPTutor: a ChatGPT-powered programming tool for code explanation*. Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky, 321–327 (2023). doi: http://doi.org/10.1007/978-3-031-36336-8_50
4. J. L. Espejel, E. H. Ettifouri, M. S. Y. Alassan, E. M. Chouham, W. Dahhane: *GPT-3.5, GPT-4, or BARD? Evaluating LLMs reasoning ability in zero-shot setting and performance boosting through prompts*. Natural Language Processing Journal, 5 (2023). doi: <https://doi.org/10.1016/j.nlp.2023.100032>
5. L. Giray: *Prompt Engineering with ChatGPT: A Guide for Academic Writers*. Annals of Biomedical Engineering, 51 (2023), 2629-2633. doi: <https://doi.org/10.1007/s10439-023-03272-4>
6. W. Holmes, I. Tuomi: *State of the art and practice in AI in education*. European Journal of Education, 57 (2022), 542–570. doi: <https://doi.org/10.1111/ejed.12533>
7. C. McGrath, T. C. Pargman, N. Juth, P. J. Palmgren: *University teachers' perceptions of responsibility and artificial intelligence in higher education - An experimental philosophical study*. Computers and Education: Artificial Intelligence, 4 (2023). doi: <https://doi.org/10.1016/j.caeai.2023.100139>
8. B. Memarián, T. Doleck: *ChatGPT in education: Methods, potentials, and limitations*. Computers in Human Behavior: Artificial Humans, Volume 1, Issue 2 (2023). doi: <https://doi.org/10.1016/j.chbah.2023.100022>
9. M. Rahman, Y. Watanobe: *ChatGPT for Education and Research: Opportunities, Threats, and Strategies*. Appl. Sci., 13, 5783 (2023). doi: <http://doi.org/10.20944/preprints202303.0473.v1>
10. M. Vukojičić, J. Krstić: *ChatGPT and AI: The Game Changer for Education* (2023). doi: <http://doi.org/10.13140/RG.2.2.31107.37923>
11. Z. Xiaoming: *ChatGPT in programming education: ChatGPT as a programming assistant* (2023).
12. R. Yilmaz, F. G. K. Yilmaz: *The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation*. Computers and Education: Artificial Intelligence, 4 (2023a). doi: <https://doi.org/10.1016/j.caeai.2023.100147>
13. R. Yilmaz, F. G. K. Yilmaz: *Augmented intelligence in programming learning: Examining student views on the use of ChatGPT for programming learning*. Computers in Human Behavior: Artificial Humans, 1, 2 (2023b), doi: <https://doi.org/10.1016/j.chbah.2023.100005>
14. B. Zhang, P. Liang, X. Zhou, A. Ahmad, M. Waseem: *Practices and Challenges of Using GitHub Copilot: An Empirical Study*. 35th International Conference on Software Engineering and Knowledge Engineering (SEKE) (2023). doi: <http://dx.doi.org/10.18293/SEKE2023-077>

„A Kulturális és Innovációs Minisztérium ÚNKP-23-4 kódszámú Új Nemzeti Kiválóság Programjának a Nemzeti Kutatási, Fejlesztési és Innovációs Alapból finanszírozott szakmai támogatásával készült.”

