TDK Dolgozat

Simon Soma Benedek

Ismételt játékok nagy nyelvi modell ágensekkel

Készítette:

Simon Soma Benedek

programtervező informatikus MSc

Témavezető:

Dr. London Anrdrás István

adjunktus

Szegedi Tudományegyetem Informatikai Intézet

Számítógépes Optimalizálás Tanszék

Szeged

2024

Tartalomjegyzék

1. fejezet

Bevezetés

A nyelvi modellek elterjedésével fontos kérdéssé vált az újabbnál újabb modellek tudásának felmérése [6], amelyre több cikk is született különböző nézőpontok alapján [2]. Elterjedt és fontos eszközzé vált az úgy nevezett „banchmarking” [14], melynek során különböző adathalmazokon tesztelik az egyes modelleket különböző feladatokra és ilyen módon hasonlítják össze a képességeiket. A Huggingface weboldalán található is egy toplista, ahol a különböző szempontok, mérőszámok alapján lehet a modellek aktuális rangsorát megtekinteni. Vannak olyan cikkek is, amelyek kódgenerálás alapján értékelték ki a nagy nyelvi modelleket [3], de mivel sok különböző feladatra alkalmazhatóak, számos területen lehet kiértékeléseket végezni, akár oktatási [8], vagy például az egészségügyi adathalmazokon is [12]. Az is egy fontos kérdés, hogy a nyelvi modellek csak olyan kérdésekre képesek-e válaszolni, amelyekről van információjuk a betanítási adatok által, vagy meg tudnak oldani komplexebb feladatokat is, amelyekhez alkalmazni kell valamilyen mélyebb tudást, nem pusztán „tárgyi” ismeretet.

Ebben a munkában azt vizsgálom, hogy az ún. nagy nyelvi modellek (LLM-ek) rendelkeznek-e ezzel a képességgel. Amikor egy forráskódot generáltatunk, az is úgymond a modellek által ismert tudásból (nyelvből) készít megoldást egy komplexebb feladatra, azonban az ilyen típusú kimenetek elemzése költséges és a kiértékelés módszertana nagyon változó lehet. Éppen ezért egy sokkal jobban definiált környezetben elemezem különböző nyelvi modellek képességeit, ami pedig nem más, mint a klasszikus játékelmélet.

A játékelmélet történelme már nagyon régre nyúlik vissza, több könyv is található róla az irodalomban, akár régebbi [5], akár újabb kiadásban [10], különböző ismeretekkel rendelkező érdeklődő számára. A különböző játékok optimális stratégiáit és ágenseit már régebb óta elemzik. A leggyakrabban úgynevezett "tournamentekkel", azaz versenyekkel [11]. Talán az egyik leghíresebb ilyen verseny az Axelrod tournament volt [9]. Mivel a játékelmélet már egy jól kiforrott eszköz a különböző játékosok képességeinek felmérésére, illetve az emberszerű gondolkodásmód vizsgálatára, így igencsak kézenfekvő eszköz, amikor egy olyan új dolgot elemzünk, mint a nagy nyelvi modellek. Éppen ezért én is az Axelrodhoz hasonló versenyhelyzetbe állítottam a nyelvi modelleket és így próbáltam mérni a teljesítményüket. A különböző nyelvi modellekkel, mint ágensek, többször ismételve játszatok bizonyos bi-mátrix, illetve kooperációs játékokat, egymás ellen, valamint néhány előre definiált játékossal szemben, azzal a céllal, hogy elemezzem a játékstílusukat, adaptációs képességüket. A kísérletek közben azt vizsgálom, hogy az egyes modellek mennyire tudják alkalmazni azokat a stratégiákat, amelyekről egyébként tudással rendelkezhetnek, illetve képesek-e felismerni bizonyos mintákat az ellenfelük stratégiáiban és amennyiben igen, kihasználják-e ezt a tudást.

Bár a téma nagyon friss, voltak már korábban, akik hasonlóképpen vizsgálták a nyelvi modelleket, ld. pl. [1, 4], azonban ezen cikkek csak a játékok egy szűk részhalmazán végez megfigyelést, nem minden típust fednek le. Így lehetnek olyan interakciók és játék szituációk, amiket nem vettek észre, vagy nem is figyeltek meg. A kooperációs játékokat illetően pedig, jelen ismereteim alapján, nem található hasonló cikk, tehát ilyen téren a dolgozatom egyedi problémát is vizsgál.

Többször lefuttatott szimulációs kísérletek után azt a következtetést lehet levonni, hogy az ismertebb mátrixjátékokra, mint a fogoly dilemma, illetve nemek harca, általánosságban jó eredményeket produkálnak az LLM-ek. Olyan esetekben azonban, amikor az egyes stratégiákat nem látjuk el beszédes nevekkel, vagy a mátrix kitalált, nem köthető direktben más ismert játékhoz, olyankor nem jó eredményeket érnek el, sokszor valamilyen nem optimális random stratégiát alkalmaznak. A többszereplős kooperációs játékok esetében az volt a megfigyelés, mintha a különböző nyelvi modellek különböző játékstílussal (személyiséggel) rendelkeznének. A Gemini gyakran módosítja a választásait, míg a GPT-3.5 inkább hamar meghozza a döntését, és azon nem módosít többet. A GPT-4 esetében az figyelhető meg, hogy ellentétben a Gemini-vel, amely csak egy bizonyos intervallumon belül módosítja választását, ha észreveszi, hogy a többi játékos nem annyira rugalmas több körön keresztül sem, ő képes kompromisszumot kötni a közös jó eredmény érdekében. Összességében a kísérletek alapján az mondható el, hogy a GPT-4 teljesít a legjobban a vizsgált modellek közül (Gemini, GPT-3.5, GPT-4) mind a mátrix játékok esetén, mind pedig a kooperációs játékoknál is. Bár néhány esetben kifejezetten rosszul teljesít ő is, messze elmarad az optimális stratégiával elérhető kifizetéstől, és nem használja fel az általa birtokolt tudást, de jobb következtetéseket von le az előző körökben nyert tapasztalatokból, illetve hajlamosabb az együttműködésre, mint a többi modell.

2. fejezet

Játékelméleti alapfogalmak és a nyelvi modellek

A játékelmélet több különböző részre bontható fel az alapján, hogy egy adott játékot hogyan lehet reprezentálni, hányan játszák, illetve milyen kimenetelei vannak az adott játékoknak. Reprezentáció alapján például tekinthetjük a kombinatorikus játékokat, amelyeket általában egy gráffal vagy hipergráffal, adott esetben fával szoktak reprezentálni. Ezzel szemben a klasszikus stratégiai játékokat inkább egy mátrix segítségével írjuk le. Különbséget lehet tenni kompetitív és kooperatív játékok között is, azaz a játékosok egymás ellen vannak, vagy egymással együttműködve tudják a számukra legkedvezőbb eredményt elérni. Ebben a fejezetben letisztázunk egy két alapfogalmat, amelyek elengedhetetlenek a dolgozat megértésében és a különböző kísérletek közötti különbség megértéséhez.

2.1. Stratégiai játékok

A legtöbb játék, amely a dolgozatban megtalálható, stratégiai játék. Ezek könnyen értelmezhetőek, illetve az ilyen típusú játékokkal egzakt felmérhető a nyelvi modellek képessége ebben az összevetésben.

Ahogy korábban már említettem, a kétszereplős stratégiai játékokat egy mátrixszal, pontosabban egy úgynevezett kifizetési mátrixszal szokták reprezentálni. Az 2.1. ábrán látható a sokak által ismert kő-papír-olló játék kifizetési mátrixa.

A játékokra több különböző csoportosítás alkalmazható, például fontos említeni, hogy

		Játékos 2		
		Kő	Papír	Olló
Játokos 1	Kő	$(0, 0)$	$(-1, 1)$	$(1, -1)$
	Papír	$(1, -1)$	$(0, 0)$	$(-1, 1)$
	Olló	$(-1, 1)$	$(1, -1)$	$(0, 0)$

2.1. ábra. Kő-papír-olló játék kifizetési mátrixa

a hagyományos kő-papír-olló teljes információs, azaz minden játékos tisztában van az összes játékos lehetséges stratégiájával, kifizetésével minden egyes játék esetén. Ezen kívül az is igaz, hogy a játék egylépéses szinkron, azaz a játékosok egyszerre választják ki a stratégiájukat a többiek döntésétől függetlenül.

Későbbiekben fontos definíció lesz számunkra a racionalitás fogalma. Ez többek között az jelenti, hogy a játékosok tisztában vannak a saját célfüggvényükkel, lehetséges stratégiájukkal és próbálják maximalizálni a célfüggvényüket, méghozzá úgy, hogy a rendelkezésre álló információik alapján a lehető legjobb döntést hozzák meg.

Fontos megkülönböztetni zérusösszegű és nem zérusösszegű játékokat. Akkor beszélünk zérusösszegű játékról, ha a játékosok összesített kifizetése minden esetben nulla. Két játékos esetén amennyi az egyik játékos kifizetése, a másik játékos pontosan annyit veszít, tehát a játékosok egymás kárára nyernek. A hagyományos kő-papír-olló egy ilyen játék hiszen, ha az egyik játékos nyer a másik veszít, döntetlen esetén pedig egyik játékos sem kap pontot, tehát a kifizetések összege minden esetben nulla. Egy játékot akkor is zérusösszegűnek tekinthetünk, amennyiben a játékosok kifizetései minden esetben egy konstans számra összegződnek, tehát nem feltétlenül nullára. Ez esetben a mátrix egy egyszerű eltolásával ekvivalens zérusösszegű játék nyerhető.

Ahhoz, hogy értelmezni/mérni tudjuk, hogy a nyelvi modell mennyire jó stratégiát alkalmaz, szükségünk van arra a háttértudásra, hogy mitől jó egy stratégia, mi az optimális lépés.

Az egyik stratégia gyengén dominálja a másikat, ha az adott stratégia az összes többi játékos összes stratégiája esetén legalább olyan jó, illetve erősen dominálja a másikat, ha szigorúan jobb, mint a másik stratégia. Egy stratégia domináns, ha az adott játékos összes többi stratégiáját dominálja.

Tiszta Nash-egyensúlynak nevezzük azt a stratégiaválasztást, amely esetén egyik játékos sem tud jobban járni, ha megváltoztatja a döntését, feltéve, hogy az összes többi játékos sem változtatja meg azt. Zérusösszegű játék esetén ez mindig egyértelmű (vagy van tisztva, vagy van kevert) és meghatározása is könnyű. Nemzérus összegű játékban is mindig van Nash-egyensúly viszont lehet egyszerre akár több, de akár nulla tiszta Nash-egyensúly is, továbbá meghatározása több szereplő esetén nem nyilvánvaló.

A stratégiai játékoknál szokták a játékokat több iteráción keresztül is futtatni és összesíteni az eredményeket. Ilyen esetekben a stratégiát úgy definiáljuk, hogy minden iterációban az előző iterációk eredményeit figyelembe véve hozzuk meg a döntésünket, hogy az adott iterációban melyik lehetséges opciót válasszuk. A kísérleteknél mi is meg fogjuk ismételni a játékokat pár körön keresztül, hogy jobban megfigyelhetőek legyenek bizonyos minták.

Példának okán a kő-papír-ollóban nincs tiszta Nash-egyensúly, hiszen gondoljuk el, hogy ha minden esetben csak az egyik opciót választjuk, az nem tűnik egy bölcs stratégiának. Ahhoz, hogy minden játékra tudjunk egy jó stratégiát definiálni szükségünk lesz arra, hogy a véletlen stratégiaválasztás is megengedjük. Az ilyen típusú stratégiákat szokták kevert stratégiának hívni. Pontosabban a játékos kevert stratégiája egy valószínűségi eloszlás a játékos által választható stratégiák között. Tehát a játékos minden választható stratégiájához egy nulla és egy közötti számot rendel, melyek összege egy, és minden körben a játékos az adott stratégiához rendelt valószínűséggel játssza az adott stratégiát. Ha a játékos az egyik stratégiáját egy valószínűséggel választja a többit pedig nulla, ebben az esetben tiszta stratégiáról beszélünk. Kevert stratégia esetén az adott játékos várható nyeresége a játékos nyereségének várható értékét tekintjük, amelyet egyszerűen ki lehet számítani az egyes játékosok valószínűségi eloszlásai és a kifizetési mátrix alapján.

A játékosok stratégiái kevert Nash-egyensúlyban vannak, amennyiben minden játékosra igaz, hogy ha módosítja a kevert stratégiáját, míg a többi játékos nem módosít azon, nem tud jobb várható értéket elérni, mint az adott kevert stratégiájával.

Az imént ismertetett fogalmakkal már könnyedén fogjuk tudni elemezni a később érintett bimátrixjátékokat, illetve a modelleket, azonban fontos ejteni pár szót a kooperációs játékok hátteréről is, hiszen a dolgozat egy része ilyen típusú játékokkal is foglalkozik, ezért a következőkben néhány ehhez kapcsolódó fogalmat ismertetek.

2.2. Kooperációs játékok

A kooperatív játékok hozzávételére azért van szükség, mivel egy szélesebb körben tesztelhetjük a nyelvi modellek következtető képességét. Az ilyen játékoknál nem minden olyan egyértelmű, mint a mátrixjátékoknál. A játékot többféleképpen lehet bemutatni, és ezáltal a modellek is másképp értelmezhetik ugyanazt a játékot.

Formailag az n személyes játék esetén adott n játékos, a játékosok közötti koalíciók halmaza, illetve az összes lehetséges koalícióhoz rendelt úgynevezett karakterisztikus függvény. A függvény megadja, hogy az adott játékoshalmazból álló koalíció esetén mekkora hasznot tud szerezni, akár a többi játékos kárára.

Az ilyen játékok esetében fontos tisztában lenni a mag fogalmával. Formálisan, egy kifizetési vektor, mely megmondja az egyes szereplők kifzetését/értékét, akkor és csak akkor van a játék magjában, ha minden koalíció számára az összes tagjának együttesen nem lehetne nagyobb kifizetése egy másik koalícióban. Más szavakkal, ha a játékosok egy csoportja nem tud többet nyerni azáltal, hogy elkülönülnek és saját koalíciójukat alkotják, akkor a játék magjában vannak. A mag léte vagy nemléte alapvető fontosságú lehet a játékelméleti elemzések során, mivel segíthet megérteni egy adott játék stabilitását és lehetséges kifizetési sémáit. Amennyiben nem a stabilitás a cél, hanem az igazságosság, akkor különböző szempontokat kell figyelembe vennünk. Az első, hogy a játékosok ereje független az elnevezésüktől, tehát bárhogyan permutáljuk őket, mindig ugyanannak az eredménynek kell kijönnie. A második, hogy a hasznot fel kell osztani, és pontosan annyit kell felosztani, amennyi a teljes haszon. A szimmetria alapján, mindenkinek a hozzátett értékével arányosan kell a kifizetés is kapnia, tehát nulla az „értéke” annak a játékosnak, akinek lényegében nincs befolyása a játék menetére. Végül, ha két független játékot játszanak ugyanazon játékosok, akkor a nyereményük a két játék összege lesz. Pontosan egy függvény van, amelyik a korábbi feltételeknek eleget tesz, még pedig a Shapley-érték.

A Shapley-érték egyedülálló tulajdonsága, hogy igazságos és effektíven elosztja a kifizetéseket a résztvevő játékosok között anélkül, hogy bármelyiküket hátrányos helyzetbe hozná. Az elv azt a koncepciót tükrözi, hogy a játékban résztvevő játékosok hozzájárulása fontos szerepet játszik a játék eredményében, és ennek megfelelően jutalmazni kell őket. A Shapley-érték kiszámítása a játékban résztvevő összes lehetséges permutáció figyelembevételével történik. Ez azt jelenti, hogy minden játékos számára megvizsgáljuk, hogy mennyi a játékosok hozzáadott értéke, amikor csatlakoznak a többi játékoshoz egy koalícióban. A Shapley-érték kiszámítása pedig az egyes játékosok által hozott „marginális" hozzájárulások átlagát veszi figyelembe az összes lehetséges játék során.

Technikaliag vesszük az összes lehetséges permutációt, ahogy a játékosok sorba rendezhetőek, majd mindegyik játékosnak vesszük a legnagyobb értéket, amellyel hozzá tudnak járulni az összeghez, az adott sorrendben. Miután kiszámítottuk az összes permutációra az egyes érétkeket, ezeket átlagolva megkaphatjuk a Shapley értéket az igazságos elosztás érdekében. Értelemszerűen a jutalmat is a fizetéssel arányosan kell elosztani.

A későbbiekben az lesz a kérdés, hogy vajon a nyelvi modell ismeri-e ezeket a számításokat, tudja-e, hogy ezeket kell használni az igazságos elosztáshoz, illetve képes-e használni ezt a tudást arra, hogy egy ilyen kooperatív játékot játszva ezt kell alkalmazni, ha valóban igazságosak szeretnénk lenni, vagy valamilyen más megoldásra jut. Még mielőtt szót ejtenénk a megoldási módról, illetve az eredményekről, érdemes egy pár szót ejteni arról, hogy mik is azok a nyelvi modellek pontosan.

2.3. Nyelvi modellek

A nyelvi modellek olyan mesterséges intelligencia (MI) rendszerek, amelyeket a természetes nyelv feldolgozására (NLP) terveztek. Feldolgozás során nem csupán az egyes szavak egyenkénti megértése a cél, hanem a szavak kontextusát is egyaránt értelmezni tudjuk. A nyelvi modelleket több különböző feladatra is lehet használni:

- **Mondatok osztályozása:** A vélemények érzékelése egy felülvizsgálat során, a spam-e az email, meghatározni, hogy egy mondat helyes-e nyelvtanilag, vagy hogy két mondat logikusan összefügg-e vagy sem stb.

- **Minden szó osztályozása egy mondatban:** A mondat grammatikai összetevőinek azonosítása (főnév, ige, melléknév), vagy a nevezett entitások (személy, hely, szervezet).

- **Szövegtartalom generálása:** Szöveg generelása adott témában, befejezetlen szöveg befejezése, hiányzó szavak kitöltése egy szövegben maszkolt szavakkal.

- **Válasz kinyerése egy szövegből:** Adott egy kérdés és egy kontextus, a kérdésre adott válasz kinyerése a kontextusban szereplő információk alapján.
- **Új mondat generálása egy bemeneti szövegből:** Egy szöveg lefordítása egy másik nyelvre, egy szöveg összefoglalása.

Ezek mellett egyéb feladatok megoldására is használható, többek között, ami számunkra fontos, párbeszédre, illetve egyéb, nem csak írott szöveges feladatoknál. Komplex kihívásokat is megold a beszédfelismerésben és a számítógépes látásban, például egy kép leírásának generálása, vagy hangmintázatok elemzése. Az ágazat fő nehézsége az információ feldolgozása, mivel a számítógép nem ugyanúgy dolgozza fel az információt, mint az ember. Az ember számára a hasonló jelentésű mondatok megállapítása egy viszonylag könnyű feladat míg ez a nyelvi modellek számára nem feltétlenül egyértelmű. Márpedig ahhoz, hogy a modell tanulni tudjon, elengedhetetlen, hogy valamilyen módon feldolgozzuk a szöveget. Mivel az egyes nyelvek bonyolultabbnál bonyolultabbak, fontos meggondolni, hogy a több módszer közül, amelyeket már kipróbáltak, adott nyelvnél adott feladatnál, milyen feldolgozási módszert érdemes alkalmazni. A dolgozat során ilyen mélységében nem kellett a modellekkel foglalkozni, mivel számunkra léteztek, már korábban elkészített modellek, amelyekkel lehetett kísérletezni. Értelemszerűen létre lehetett volna hozni egy új modellt, amely kifejezetten ilyen játékelméleti feladatok megértésére és megoldására van optimalizálni, azonban a dolgozat célja nem ez volt, hanem a már meglévő nagyobb, híresebb generatív modellek tesztelése.

Mint azt korábban már említettem, ezek a modelleket általában valamilyen speciális feladat megoldására lettek kitalálva. Sok cikk foglalkozik azzal a feladattal, hogy a különböző nyelvi modelleket osztályozza, kiértékelje [13]. Nagyon nehéz olyan modellt találni, amely minden téren megállja a helyét. Éppen ezért a kísérletek során minél több fajta modell tesztelése volt a cél, hogy lássuk, hogy a nagyobb nyelvi modellek közül melyek lehetnek azok, amelyek ilyen játékok játszására jelenleg alkalmasak lehetnek és képesek akár megtaláni az optimális stratégiákat is.

3. fejezet

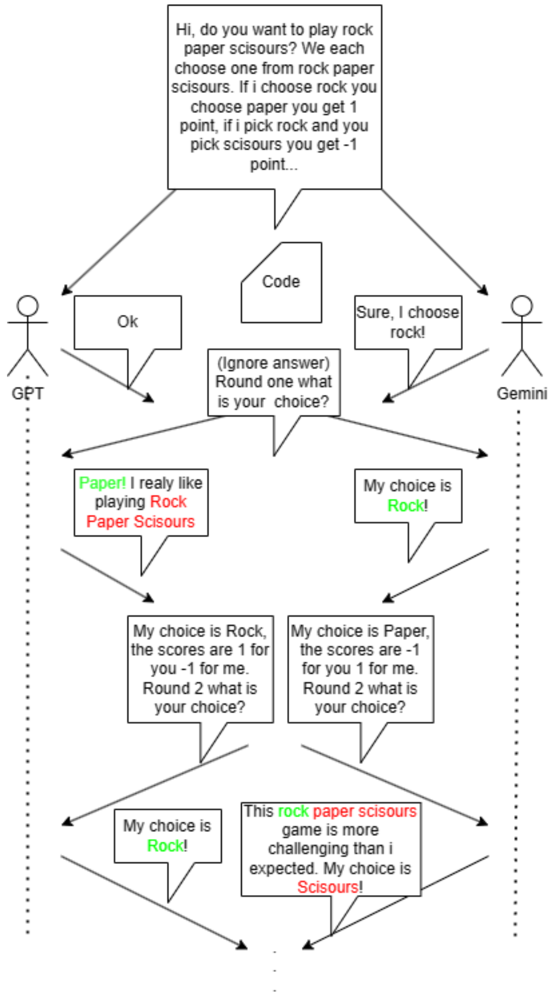
Megközelítés

3.1. Felépítés

A kísérletek futtatása kötött keretek között zajlik. Eleinte voltak próbálkozások arra, hogy a nyelvi modellek a játékot elindító szöveg után már önmaguktól kommunikáljanak egymással, azonban ezek a beszélgetések minden esetben elmentek más irányba és sosem tudtak eljutni tíz körig. Ezen tapasztalatok alapján végül nekem kellett az egyes üzeneteket feldolgoznom és átadnom a másik modellnek egy jól megfogalmazott környezetben, hogy a kísérletek jól végig fussanak.

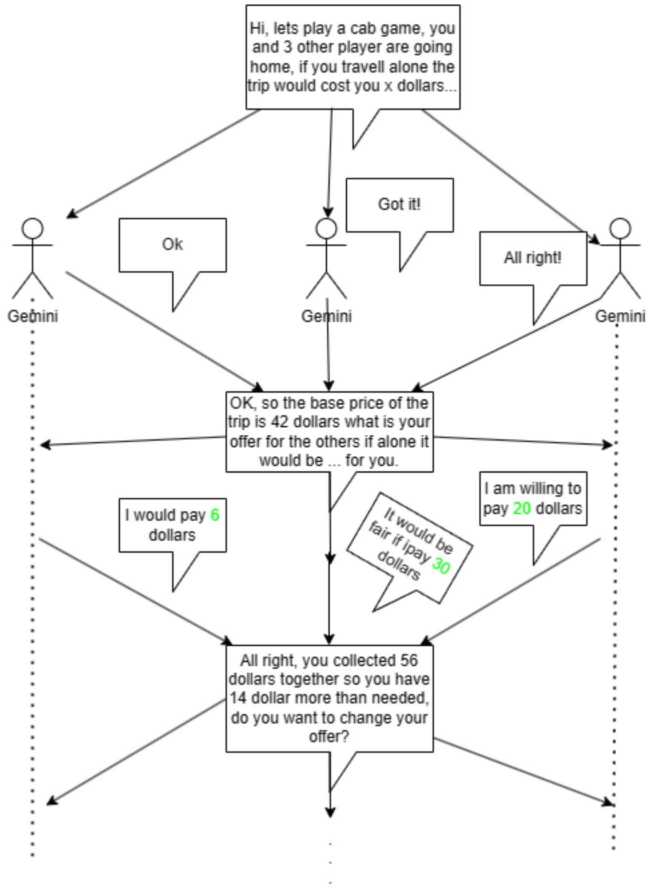
Ahogy a 3.1. ábrán is látszik, első körben ismertetem a játékot a modellekkel, jelen esetben a kő-papír-ollót. A játékot szövegként írjuk le, tehát mindkét játékosnak elküldjük, hogy „ha te ezt választod, az ellenfeled ezt választja, akkor te ennyi pontot kapsz, ő ennyi pontot kap...". A stratégiák minden esetben értelmes diszkrét szavak, mivel ezt egyszerűbb kezelni, amikor a válaszokat feldolgozzuk. Ezen kívül a kezdő szövegben felhívjuk a figyelmüket a modelleknek, hogy csak egy szóval válaszoljanak, illetve amikor nem egy szóval válaszolnak, a legelső olyan szót tekintem a válaszának, amelyik egyezik valamelyik stratégia nevére kisbetűssé téve. Tehát az első üzenet után minden körben, ha volt már korábbi kör, akkor elküldöm az előző kör eredményét és az eddigi játékállást, majd megkérdezem, hogy mi a választása a következő körre. Az adott válaszból kiolvasom a megfelelő stringet, majd frissítem a játékállást és a következő kör következik.

A kooperációs játékok esetében az adat áramlás nagyjából hasonló, a játék felépítésében illetve a játékosszámban van különbség.



3.1. ábra. Játékosok kommunikációjána kezelése

A 3.2. ábrán is látszik, hogy a kezdőszöveg teljesen hasonló, csak egy másik játékot ír le, illetve, hogy hogyan fog lezajlani egy játék. Az első kör után megkeressük az első számot az egyes modellek által adott válaszokban regex segítségével, ezeket az értékeket lementjük, majd összeadjuk, hogy összességében mennyire vannak távol csoportosan a játékosok kitűzött céltól. Ezek alapján a következő körben elküldjük a játékosoknak, hogy ki mennyit ajánlott, mennyi pénzt gyűjtöttek össze együtt és mennyire van még szükség. Ezek után pedig megkérdezzük, hogy akar-e módosítani az általa felajánlott összegen, vagy nem módosít. A megegyezésre tíz kör áll rendelkezésre, ha tíz kör alatt sem egyeznek meg akkor is vége a játéknak. Akkor tekintjük úgy, hogy a játékosok megegyeztek, ha játékosok által adott összeg pontosan kiadja a szükséges mennyiséget, és már egy kör óta egyik játékos se módosított az általa megadott értéken.



3.2. ábra. Játékosok kommunikációjána kezelése kooperációs játéknál

3.2. Ágensek

Értelemszerűen ahhoz, hogy olyan eredményeket kapjunk, amelyből jól lehet megfigyeléseket tenni, nem elegendő csak nyelvi modelleket alkalmazni a kísérlet során.

Éppen ezért készítettem egy random játékost, amelynek inicializáláskor át kell adni a lehetséges opciókat, az pedig a random csomag segítségével véletlenszerűen választ egyet az opciók közül. Ez az ágens arra jó, hogy egyes játékoknál, ahol ez az optimális stratégia, mint a kő-papír-olló, összevethessük vele a modelljeinket.

Készült egy "egyenes" játékos is, amely inicializáláskor megkapja, hogy melyik választási lehetőséget játssza, és minden egyes körben csak visszaadja azt, amit inicializáláskor kapott. Szintén van játék, ahol ez az optimális, valamint olyan szempontból is hasznos, hogy egy nagyon szélsőséges játékos és meg lehet vizsgálni, hogy a modellek felismerik-e azt, hogy ők nem váltanak stratégiát. Ezen felül az ábrázolás fejlesztésénél a tesztelést segítik, hogy az ábrák jól vannak-e leimplementálva.

Talán az egyik legbonyolultabb „alap” játékos a kevert játékos, amely megkapja a választási lehetőségeket, illetve a valószínűségeket az egyes választásokhoz, és ezek alapján az előre beállított kevert stratégiát játssza. Igazából jól beállítva ezzel a modellel a másik két játékost is szimulálni lehet, hiszen a valószínűséget egyre állítjuk valamely választásra akkor elérjük, hogy mindig azt válassza, ha meg egyenlően elosztjuk akkor pedig a random játékos reprodukálható. Amellett, hogy van olyan játék, amelyben csak optimális kevert stratégia létezik, ezek az ágensek nagy segítséget nyújtottak a diagrammok előállításakor, például amikor azt vizsgáltam, hogy az egyes játékosok milyen eloszlásban válasszák az egyes opciókat a kevert stratégiát használó játékossal jól ellenőrizni lehetett a függvény működését.

Kifejezetten a fogolydilemma szerű játékokra készült még egy TitForTat játékos is, amelynek az a stratégiája, hogy az első körben kooperál, a következő körökben pedig az ellenfele előző stratégiáit játssza. Ez az ágens természetesen könnyedén kijátszható, ha tudjuk, hogy róla van szó, de korántsem biztos, hogy a nyelvi modellek ezt felismerik, éppen ezért egy érdekes ellenfél lehet.

A nyelvi modellek közül többet is vizsgáltam, azonban a végső kísérletekbe csak három került be.

A ChatGPT esetében szükség volt az OpenAI csomagra, mert ennek segítségével lehetett API hívásokat kezdeményezni az OpenAI felé. Példányosítani kellett az OpenAI objektumot, méghozzá egy úgynevezett API kulccsal. Az API kulcs szolgált arra, hogy hozzákösse a hívást az OpenAI fiókhoz, és ezáltal tud számlázni, hiszen a ChatGPT használata hívásonként fizetős, tehát használat előtt pénzt is kellett feltölteni. Használat közben az üzeneteket menedzselni kellett, hiszen az OpenAI ez alapján ismerte a kontextust, új játékos esetén pedig ezt a kontextust törölni kellett. A modell önmagában szöveg generálására használható, azonban függvény segítségével beszélgetést is lehet létrehozni, aminek segítségével lehet a kontextust módosítani. A függvény hívásakor meg lehet adni, hogy melyik modellel szeretnénk létrehozni a chatet, ezt az ágens inicializálásánál állítottam be, így egyszerűen lehetett GPT3.5 és GPT4-is ágenst létrehozni. Utóbbi esetében az input és output tokenek ára drágább, de elméletben, korábbi tapasztalatok alapján szinte minden területen jobb eredményeket produkál. Fontos, hogy az üzeneteket felcímkézzük szerepkör és tartalom alapján, mert más különben a modell nem tudja feldolgozni az inputot.

Természetesen a Gemininek is készült egy csomagja, amelyet be kellett importálni, illetve kellett egy API kulcs a ChatGpt-hez hasonlóan. Sajnos a Gemini API európában nem elérhető, ezért szükség volt VPN-re, hogy egyáltalán API kulcsot létre lehessen hozni. Miután az API kulcs bekerült a Colab környezetbe, már nagyon egyszerű volt a használata. Létre kellett hozni a modellt, amelyhez a chatet lehetett incializálni, majd ahhoz lehetett üzeneteket küldeni. Készítettem egy szimpla ciklus az üzenet feldolgozása részhez, arra az esetre, ha küldés közben valami internetkapcsolati, vagy egyéb hiba lép fel. Ebben az esetben megismételjük a kérést. Néhányszor előfordult tesztelés közben és kellemetlen tud lenni, amikor egy hosszú futtatás végén előjön egy random hiba, ami miatt az egész futtatás megáll.

Emellett próbálkoztam egyéb modellekkel is a Huggingface könyvtárból (pld: Llama, Falcon, Microsoft DialogGPT, Phi-2), azonban a legtöbb modell nem rendelkezett annyi paraméterszámmal, hogy értelmes választ adjon még egy egyszerűbb kérdésre is, vagy nem olyan adatokon tanították be, hogy erre alkalmas legyen (például leginkább orvosi beszélgetéseken betanított az egyik ilyen modell), ezért inkább GPT-3.5, GPT-4, valamint a Gemini-vel folytattam a teszteket.

3.3. Kiértékelés

Készítettem több fajta diagrammot azzal a céllal, hogy a modellek különböző tulajdonságaira, játékstílusára világítsanak rá és később ezek alapján lehessen megfigyeléseket tenni a 4. eredmények fejezetben.

Az alábbi függvényeket írtam meg a kiértékeléshez:

```
experiment.visualize_scores_matrix()
experiment.visualize_normalized_scores()
experiment.plot_win_rates()
experiment.plot_choice_preferences()
experiment.visualize_choice_modification('r1', 'r2', 0, 4)
experiment.average_aggregate():
```

Az első a játékosok adott ellenfelek ellen játszott játékait úgy rajzolja ki egy mátrixban, hogy az adott sorban található játékos mennyi pontot szerzett az oszlopban található játékos ellen. Ezen a diagrammon nagyon jól látszódik például a zéruösszegű játékok esetében, hogy valóban annyit nyer az egyik játékos, mint amennyit a másik veszít.

A második a korábban említetthez hasonló, de az egész versenyben szerzett összesített pontszámokat és elérhető pontszámok hányadosát adja meg az egyes játékosokra, ezáltal egy százalékos összteljesítményt kapunk.

A harmadik mátrixban mutatja az egyes játékosok elleni győzelmek vereségek és döntetlenek számát. Ennek a mátrixnak nem minden esetben van értelme, például a kő-papírollónál releváns lehet, de ha a kifizetési mátrix olyan, hogy az egyik játékos minden esetben több pontot szerez, mint a másik, akkor semmiképpen nem tud több pontot szerezni ellenfelénél, ezért a relatív teljesítmény, mint ahogyan az előző diagrammon volt ábrázolva, relevánsabb információ lehet.

A negyedik diagram a választási preferenciákat mutatja, melyik játékos melyik stratégiát hányszor választotta. Ez azért jó, mert egyes játékok esetében mindenféleképpen valami kevert stratégiát érdemes játszani, egyesesetekben pedig csak egy fajta stratégia játszása az optimális, és ezáltal meg lehet vizsgálni, hogy az egyes játékosok milyen eloszlásban választották az egyes stratégiákat.

Az ötödik függvény egy olyan diagramot rajzol ki, amely két adott játékos közötti választás módosításokat vizualizál, adott játékokban indexek alapján. Például a kódrészletben látható függvényhívás a két random választó ágens elsőtől az ötödik játékág bezárólag levő választásait és annak módosításait ábrázolja. Ez a fajta ábrázolás azért jó, mert az olvasható le róla, hogy a játékosok hogyan reagálnak az ellenfeleik stratégiáira, vagy azok módosítására.

Az utolsó diagram pedig egy olyan ábra, amely az aggregált átlagot valamint szórást ábrázolja az egyes játékosokra. Ez az ábra azt segít belátni, hogy mekkora a várható kifizetése egy adott játékosnak, illetve milyen tartományban mozoghat. Tesztelés közben például a kő papír olló játéknál megfigyelhető volt az az érdekes összefüggés, hogy amikor egy szimpla tiszta stratégiát játszó (tehát például minden esetben követ választó) játékos játszik egy vagy több random stratégiát játszó játékossal, akkor az átlagos kifizetése hasonló lesz az optimális kevert stratégiát használó játékosokéhoz (mivel a random játékos nem realizálja, hogy az ellenfele nem módosít stratégiát több körön keresztül, ezáltal néha nyer néha veszít), azonban a szórása sokkal nagyobb.

4. fejezet

Játékok, eredmények

A játékok bemutatása a kisebb mátrixtól a nagyobb fele, egyszerűbb játékoktól a bonyolultabb fele történik. Előbb a zérusösszegű, majd a nem zérus összegű, legvégül pedig a kooperációs játékról lesz szó. A játékokról először egy általános leírást adok, majd pedig a nyelvi modellek által elért eredményeket tárgyalom. Ha az olvasó jobban elmélyedne a játékokokban, (azon játékok kivételével, amelyeket én találtam ki) a [7] könyv egy jó választás.

4.1. Zérusösszegű játékok

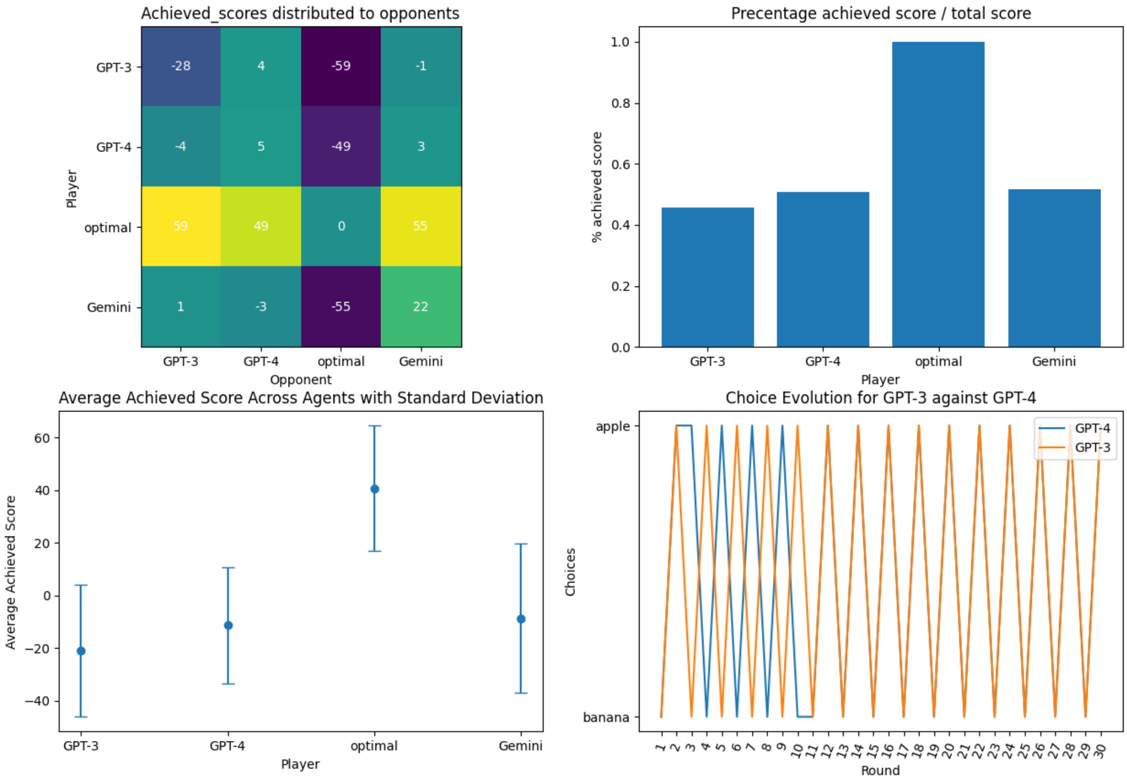
4.1.1. Példa játék tiszta Nash egyensúllyal

A következő 4.1. ábrán látható mátrixszal definiált játéknak nincs kitüntetett neve. A mátrix értékeket magam kreáltam úgy, hogy bizonyos kritériumoknak megfeleljen és egy nagyon egyszerű első teszt játék legyen.

	Döntés 1	Döntés 2
Döntés 1	$(0, 0)$	$(-1, 1)$
Döntés 2	$(1, -1)$	$(0, 0)$

4.1. ábra. Példa 2x2 játék tiszta Nash-egyensúllyal

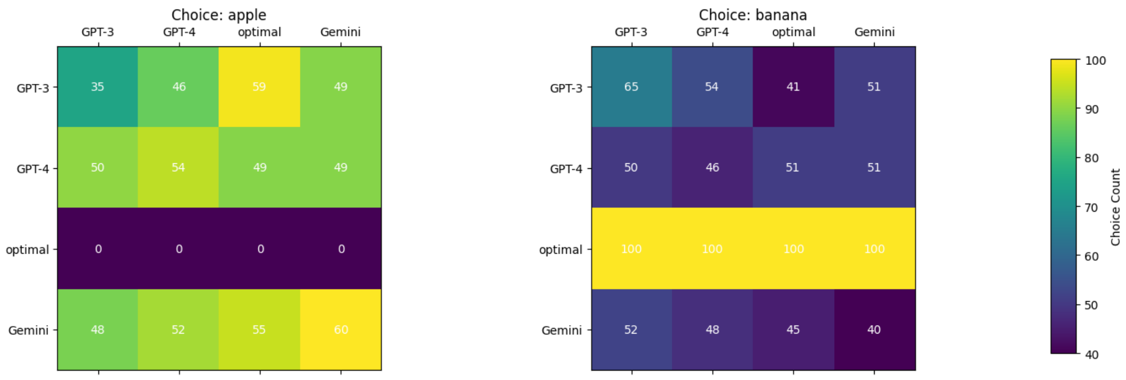
Mint az a mátrixon is egyértelműen leolvasható, a játék zérusösszegű, hiszen a játékosok kifizetései nullára összegződnek minden esetben. Emellett a játéknak tiszta Nash egyensúlya van. A Döntés 2 esetén egyik játékosnak sem éri meg stratégiát váltani, hiszen ha váltanának, negatívvá változna a kifizetésük, a másik játékos pedig jobban járna. Érdekes, hogy bár a Döntés 1 ugyanezekkel a kifizetésekkel rendelkezik, abban az esetben mind a két játékosnak érdekében áll módosítania a stratégiáján, hiszen nőne a kifizetésük, tehát az nem lehet egyensúly. Ugyanerre a konklúzióra úgy is rájöhetünk, amennyiben a dominanciát vizsgáljuk. Az első játékos esetén a második sor dominálja az elsőt, tehát neki csak a második sort érdemes választania, hiszen akkor minden esetben nagyobb a kifizetése. A második játékos számára szintén a második stratégia dominálja az elsőt, tehát egy racionális játékos csakis a másodikat játszaná, ebből következik, hogy elméletileg minden körben a Döntés 2 választásnak kellene bekövetkeznie.



4.2. ábra. Eredmények a 2x2 tiszta Nash-egyensúllyal rendelkező játékhoz

A 4.2. ábrán látható, hogy az optimális játékos szinte dominálja a többi játékost, ami várható volt, de nem ekkora mértékben. Az első diagrammon látszik, hogy általában 100 pont különbség van az ő és az ellenfelei pontjai között (-50, 50, zérus összegű játék miatt). Ebből kikövetkeztethető, hogy az ellenfelek nagyjából a 100 lejátszott körből 50-

szer választották a másik opciót, tehát fele-fele arányban játszották az egyes stratégiákat. Ugyanez látszódik a 4.3. ábrán is.



4.3. ábra. Egyes játékosok stratégiáinak eloszlása a 2x2 tiszta Nash-egyensúllyal rendelkező játékhoz

A 4.2. ábrán a 2. és 3. diagram szintén azt mutatja, hogy az optimális stratégiát játszó játékos sokkal jobban teljesít átlagosan, illetve a többi játékos várható kifizetése nagyjából ugyanott mozog, hiszen ugyanúgy random választanak stratégiát. Ezek váltakozása körről körre a 4. diagrammon látható éppen a két GPT modell esetében. Abban az esetben amikor átfedés van a két játékos stratégiáinak módosítása között, csak egy szín jelenik meg, tehát majdnem, hogy teljesen ugyanúgy változtatták a stratégiájukat az egyes játékosok ebben az esetben.

Az eddigi tesztek alapján az a következtetés vonható le, hogy ha egy ismeretlen játékot próbálunk megtanítani a modelleknek egy kifizetési mátrix segítségével, akkor nem próbálják megérteni, hanem random stratégiát választanak. A következőkben olyan játékok következnek, ahol a random választás az optimális és gondolhatnánk, hogy a nyelvi modellek ezen esetekben jól fognak teljesíteni.

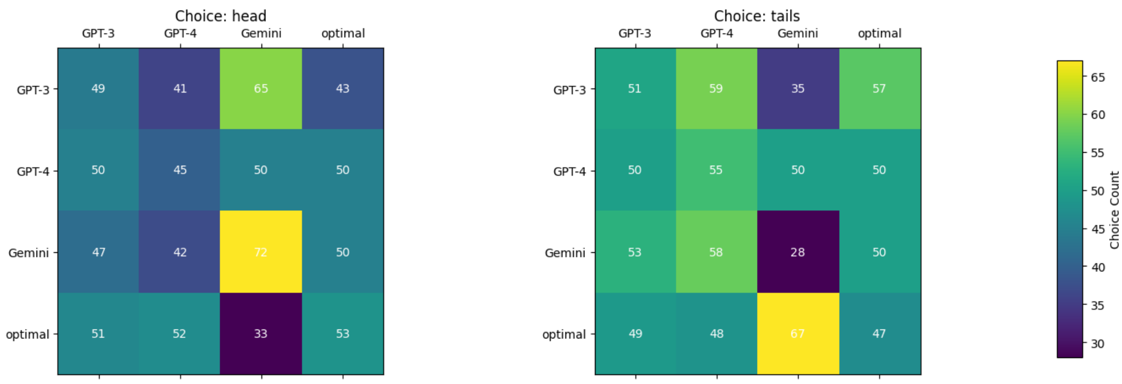
4.1.2. Fej vagy írás

A 4.4. ábrán látható mátrixszal definiált játék szintén nagyon egyszerű, azonban nincs benne tiszta Nash egyensúly. A játékosok választanak, hogy fej vagy írás és amennyiben ugyanazt tippelték a sorjátékos nyer, ha különbözőt, akkor a másik, azaz oszlopjátékos. Olyan, mintha az egyes játékos érmét dobálna, azonban ebben az esetben a második játékos az érme és nem feltétlenül 50% valószínűséggel lesz egyik vagy másik a kimenetel.

	Fej	Írás
Fej	$(1,-1)$	$(-1,1)$
Írás	$(-1,1)$	$(1,-1)$

4.4. ábra. Fej vagy írás játék kifizetési mátrixa

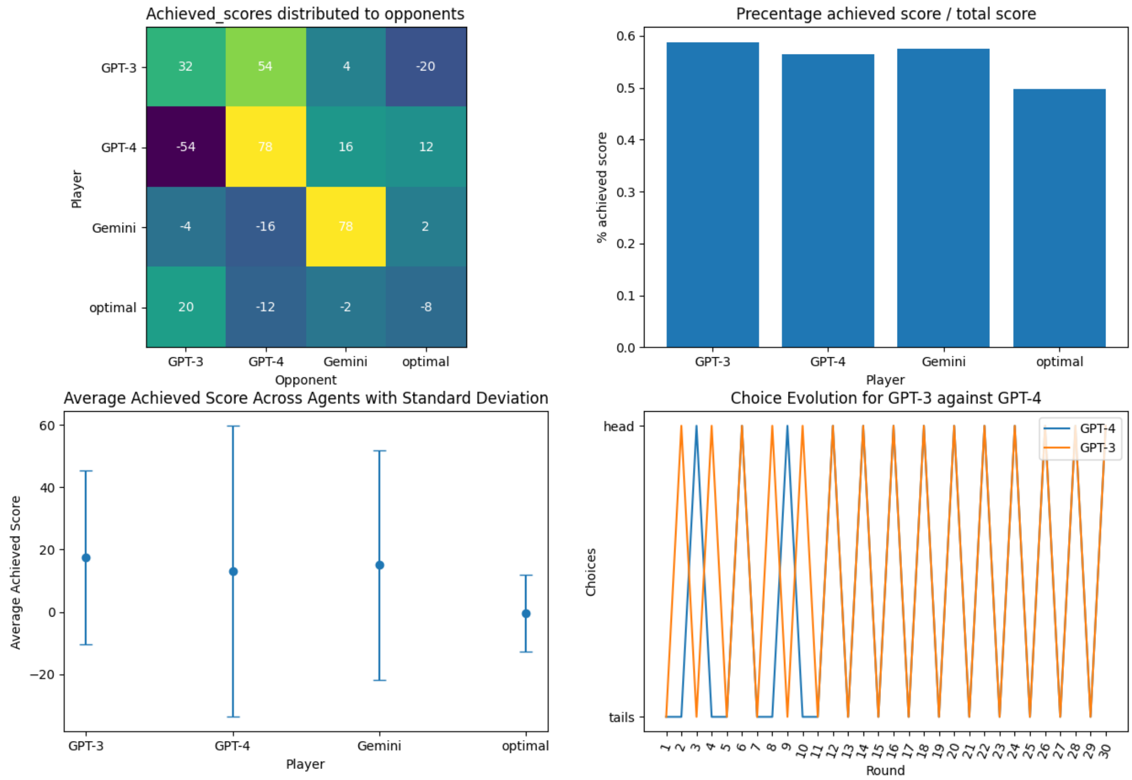
A játék szintén zérusösszegű, mint az előző, hiszen amennyit az egyik játékos nyer, annyit veszít a másik, azonban ebben a játékban nem található domináns stratégia egyik játékos számára sem, eltérően az előző játéktól. Tiszta Nash egyensúly sincs, hiszen minden esetben valamelyik játékosnak megéri stratégiát váltani. Az 1-1 és 2-2 választás esetén a kettes játékosnak, az 1-2, 2-1 választás esetén pedig az egyes játékosnak éri meg stratégiát vállani. A kevert stratégiákat tekintve, egész egyszerűen kikövetkeztethető, hogy a stratégiákat fele-fele arányban érdemes választani mindkét játékosnak, hiszen a kifizetések megegyeznek mindkét stratégiánál, és a várható érték 0 lesz.



4.5. ábra. Fej vagy írás stratégiák eloszlása az egyes játékosokra

A 4.5. ábra alapján valóban arra kövtkeztethetünk, hogy a modellek általában random választanak, mivel nagyjából egyenlő az egyes stratégiák eloszlása.

A 4.6. ábrán pedig az látszik, hogy valóban a kevert stratégiával lehet a legjobb eredményeket elérni. Megfigyelhető az is az egyes játékosok elleni stratégiamódosításokat vizsgálva, hogy a GPT-4 próbálja ráilleszteni az ő stratégiáját az ellenfelére, ami nem is olyan rosszul sikerül neki több játékost is figyelembe véve. Attól függően, hogy a GPT-4



4.6. ábra. Fej vagy írás eredmények

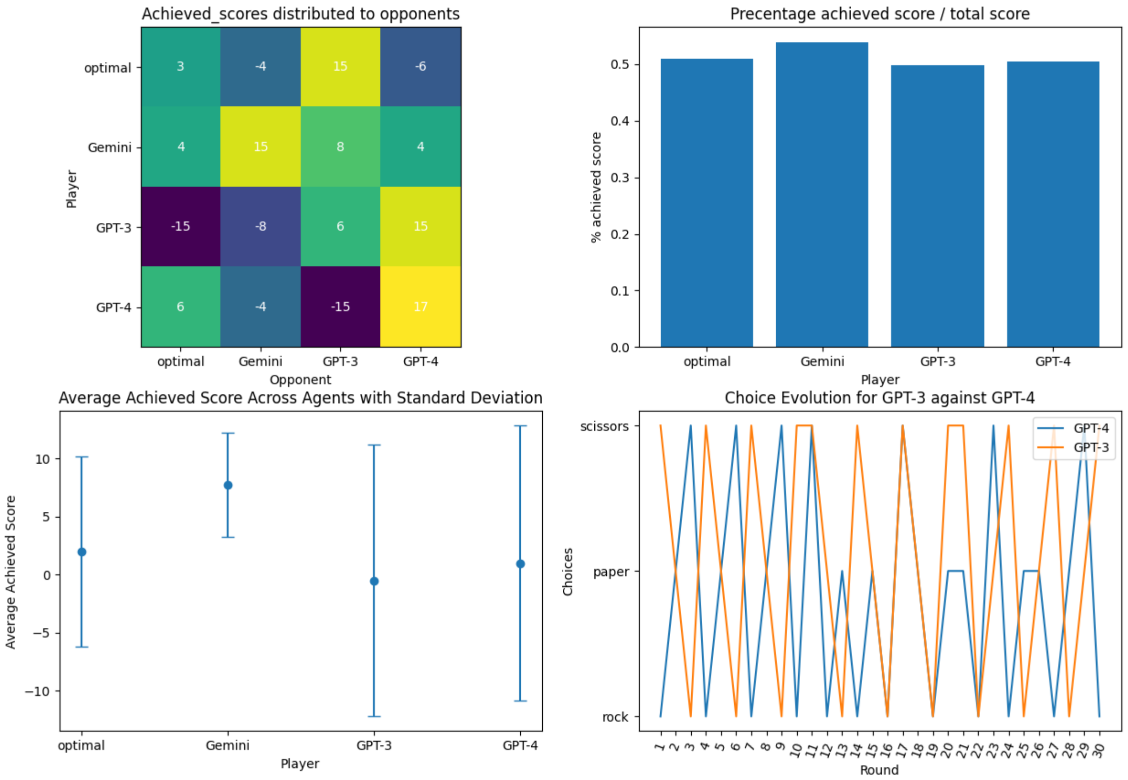
éppen 1-es vagy 2-es játékos, ezzel a stratégiával elég szélsőséges pontokat tud szerezni, éppen ezért olyan nagy a szórása a harmadik diagramon.

4.1.3. Kő-papír-olló

A hagyományos kő papír olló játékról (2.1. ábra) már korábban volt szó, az előző játékhoz nagyon hasonló, csak 2x2-es mátrix helyett 3x3-mas. Nincs benne Nash egyensúly vagy domináns stratégia. Az egyetlen optimális stratégia, ha azonos valószínűséggel (1/3) választjuk az egyes stratégiákat.

A 4.7. ábrán látottak alapján elmondható, hogy a Kő-Papír-Olló esetében olyan eredményeket kaptunk, ami várható volt. A modellek nagyjából random választottak (lásd 4.8. ábra), ezáltal a várható kifizetésük is 0 felé közelített, a kisebb kiugrások az egyes modellek által csak a véletlenek sorozatának tudható be.

Nagyjából kezd körvonalazódni, hogy a nyelvi modellek ténylegesen csak random választanak stratégiát és ezáltal az ilyen játékokat leszámítva nem túl nehéz ellenfelek a



4.7. ábra. Kő-Papír-Olló eredmények

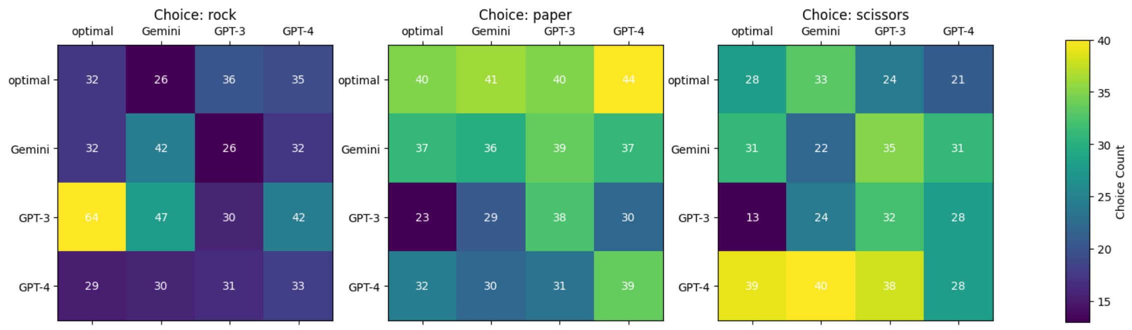
mátrixjátékokban. Azonban mielőtt ezek alapján a kísérletek alapján levonnánk a konklúziót, próbáljunk ki még néhány típusú játékot, a biztos eredmények érdekében.

4.2. Nem zérusösszegű játékok

4.2.1. Fogolydilemma

A fogolydilemma egy népszerű játékelméleti példa, amely a vádalku intézményére épül. A rendőrség letartóztat két bűnözőt (2 játékos), azonban a bűntényre nincsen tárgyi bizonyíték, tehát vallomásra kell bírni őket. Elszigetelik őket egymástól, hogy ne tudjanak kommunikálni. Ha egyik játékos sem vall, úgy elenyésző büntetést kapnak bizonyíték hiányában. Amennyiben az egyik vall a másik tagad, úgy amelyik vallott nem kap büntetést, ellentétben a másik játékossal, aki az eredeti büntetés többszörösét kapja, viszont, ha mind a kettő vall, akkor mind a kettő magas büntetésre számíthat.

Ahogy a 4.9. ábrán is látható a mátrix szimmetrikus és nem zérus összegű. A kifizetési



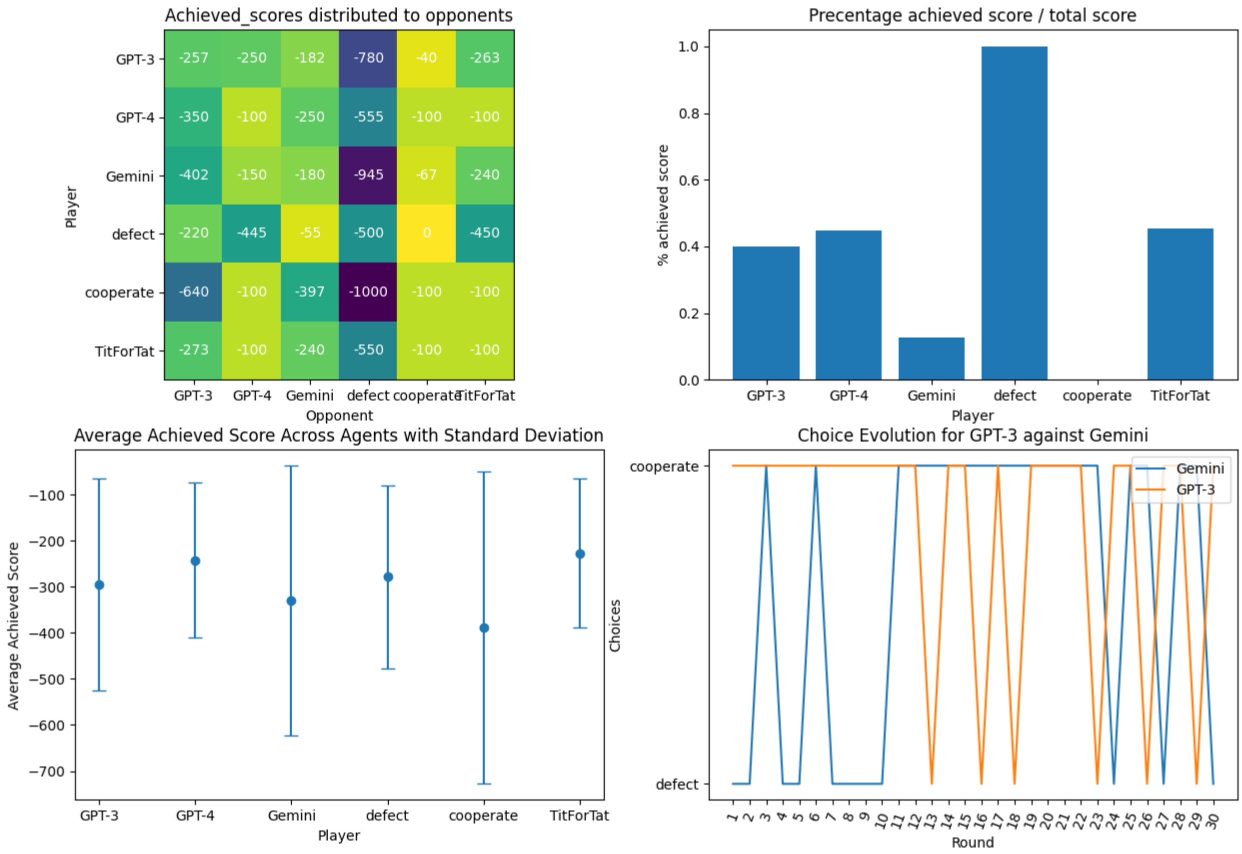
4.8. ábra. Kő-Papír-Olló stratégiáinak eloszlása az egyes játékosok által

	Vall	Tagad
Vall	$(-6,-6)$	$(0,-10)$
Tagad	$(-10,0)$	$(-1,-1)$

4.9. ábra. Fogolydilemma kifizetési mátrixa.

mátrixot egyébként más számokkal is szokták definiálni, a lényeg az, hogy az egyes cellák értékeinek egymáshoz viszonyított előjele ne változzon. Például lehet szó kevesebb börtönbüntetésről is a bűnténytől függően, de a játék ugyanaz marad. Ha elkezdjük vizsgálni a mátrixot, hamar rájöhetünk, hogy mindkét játékos számára domináns stratégia az, ha vall. Hiába lenne egy sokkal jobb kifizetés, mindkettőjük számára azzal, ha mind a kettő tagad, mivel el vannak különítve, nem tudják megbeszélni egymással a stratégiájukat és a másik stratégia kifizetése minden esetben mind két játékos számára jobb.

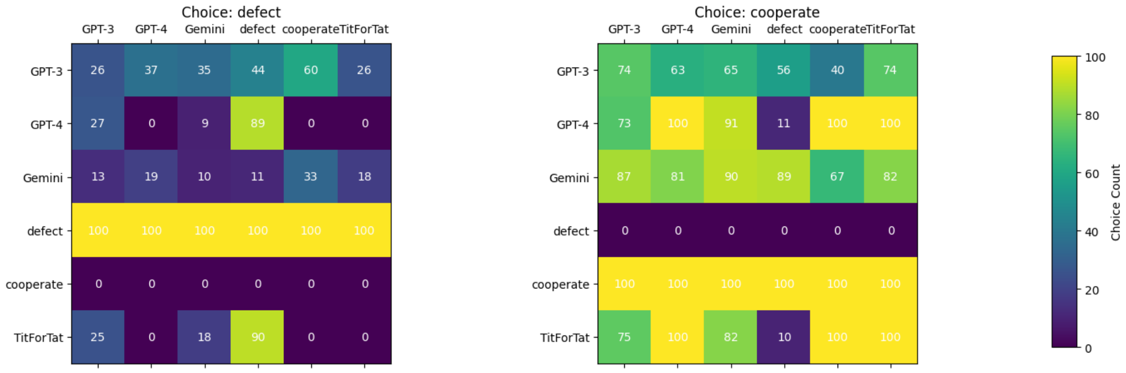
Ennél a játéknál érdekes lehet egy olyan játékos, amely az ellenfele korábbi választásaira reagál, éppen ezért itt létrehoztam egy TitForTat nevű játékost, amelyről korábban már volt szó. Emlékeztetés képpen ez a játékos az első körben kooperál, majd az összes további körben az ellenfele stratégiáját játssza. Ez nem tűnhet egy optimális játékosnak, azonban annál emberibb mivel, ha az ellenfele nem vall ő sem fog. Ami látszik a 4.10. ábrán és nagyon érdekes, hogy bár elvileg a folyamatos defektálás azaz vallás a nyerő stratégia, és talán egy lájátszott kör esetén azt is érdemes választani, ismételt játék esetén, ha a játékosok hajlandóak néha kooperálni, akkor végeredményben sokkal jobb eredményt tudnak elérni. Hiába látható a második diagramon, hogy a folyamatosan valló játékos az általa elérhető maximális pontot elérte, a nyelvi modellek főleg a GPT-4 képes



4.10. ábra. Fogoly dilemma eredmények

volt felismerni a mintázatot az ellenfele stratégiájában, ezáltal nem, vagy csak nagyon ritkán próbált vele kooperálni. Éppen ezért a harmadik diagrammon is látszódik, hogy átlagosan a GPT-4 és a TitForTat játékos még több pontot is szerzett, mint az egy kör esetén optimális stratégia. Amennyiben egyetlen játékos se lett volna, amelyik hajlandó lett volna kooperálni, valószínűleg a tiszta vall stratégia nyert volna, ezért is olyan érdekes ez a példa.

Több érdekes diagram készült ehhez a kísérlethez, de talán az egyik legbeszédesebb a 4.11. ábrán látható. Itt is látszik, hogy a GPT-4 mind a TitForTat, mind pedig a csak kooperáló játékossal végig kooperált, ezáltal elég sok „pontra", illetve kevesebb negatív pontra tettek szert. Ezzel ellentétben a folyamatosan defektáló játékossal szemben csak defektált a modell. Ez alapján az ábra alapján lehet arra következtetni, hogy mégsem szimpla random választás alapján választott stratégiát, hanem valamilyen szinten felismeri az ellenfele stratégiáját, és hogy hogyan kellene neki is választania. Az viszont



4.11. ábra. Fogolydilemma stratégiák eloszlása

kérdéses, hogy ha ez a modell ennyire jól tudja értelmezni a feladatot és az ellenfele stratégiáját, akkor az első kísérlet esetén, ahol nyilvánvaló volt, hogy mi az optimális stratégia, miért nem próbálkozott annak játszásával?

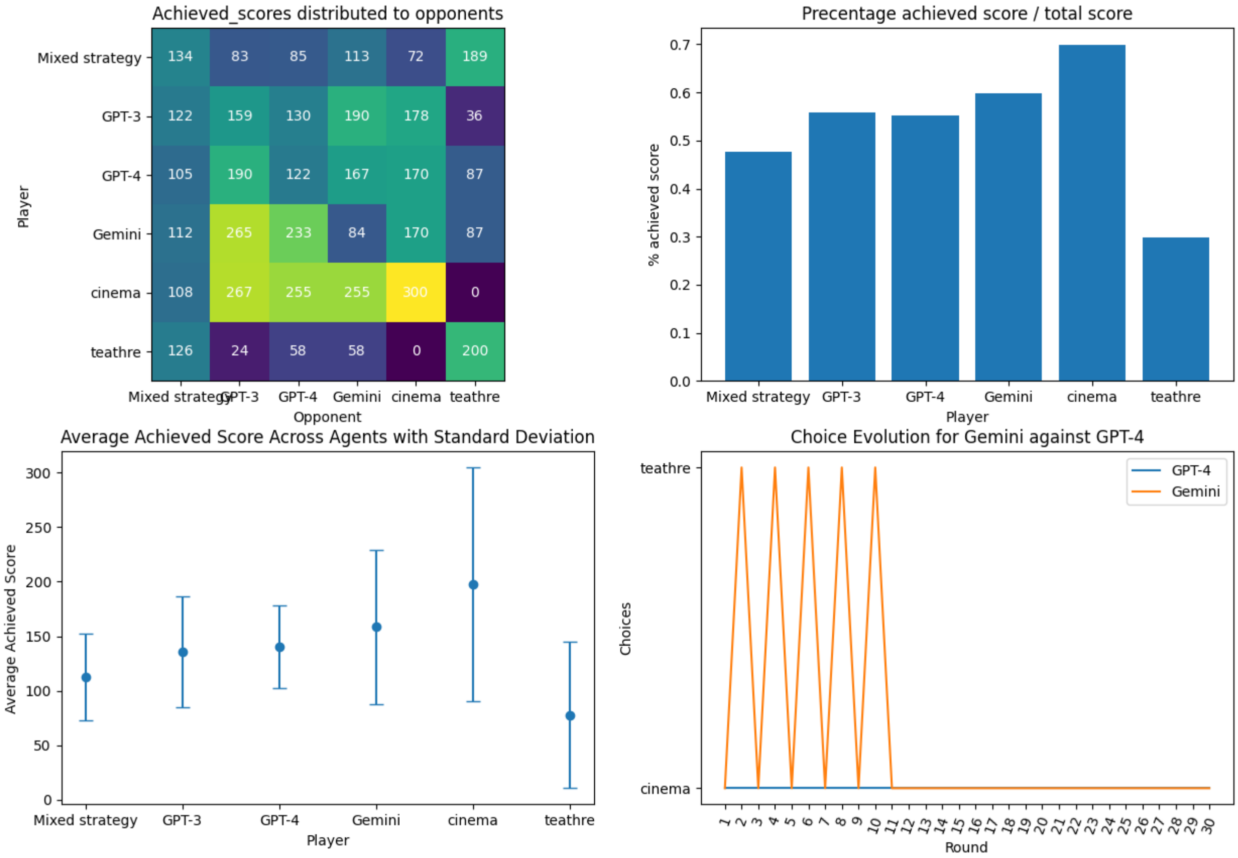
4.2.2. Nemek harca

A nemek harca játékban a két játékos, egy fiú és egy lány szeretné eldönteni, hogy hogyan töltsék a szabadidejüket. Mind a két játékosnak megvannak a preferenciái, azonban inkább közösen mennek valahova, ahova nem szeretnének, mint külön-külön oda, ahova szeretnének.

	Mozi	Focimeccs
Mozi	$(2, 3)$	$(0, 0)$
Focimeccs	$(0, 0)$	$(3, 2)$

4.12. ábra. Nemek harca játék kifizetési mátrixa.

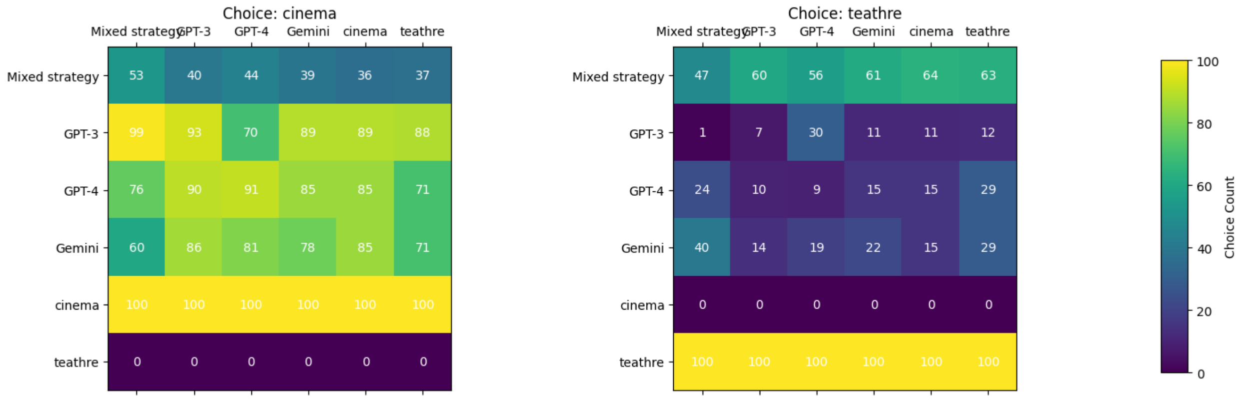
A fenti 4.12. ábra egy példa, de a stratégiák nevei értelemszerűen módosíthatóak, illetve az értékek megfelelő módosításaival is lehet kapni hasonló példát. Nincs domináns stratégia egyik játékos számára sem, azonban van két tiszta Nash egyensúly is. Amennyiben mind a kettő a mozit, vagy mind a kettő a focimeccset választja, egyik játékosnak sem érdemes módosítania a választásán, hiszen csak csökkenne a kifizetése. Van egy optimális kevert stratégia is, méghozzá az, hogy mind a két játékos 60% valószínűséggel választja azt, amelyiket ő jobban preferálja és 40% valószínűséggel azt, amelyiket nem. Ebben az esetben mindkét játékos várható kifizetése 1.2 lesz.



4.13. ábra. Nemek Harca eredmények

Az eredményeket tekintve (lásd 4.13. ábra) a modellek egész jól játszák a nemek harca játékot. Az alapján, hogy a modellek sor vagy oszlopjátékosok éppen az jött ki, hogy inkább a mozit választják általánosságban mivel az számukra több pontot ér, azonban egy másik permutációban, lehet, hogy a színházat választották volna gyakrabban. Összességében, ha a stratégia módosításokat tekintjük, jellemző rájuk, hogy egy idő után elkezdik az ellenfelük stratégiáját választani és kooperálnak a több pont érdekében ezért a teljesítményük elég jó, főleg a kevert stratégiájú játékossal szemben.

Sajnálatos módon a 4.14. ábrán az is látszik, hogy ha csak a színházat játszó játékost tekintjük, egyik nyelvi modell sem hajlandó csak azt választani, amikor ő az ellenfelük. Tehát a modellek nem annyira figyelnek az ellenfelük stratégiáira, mint a fogolydilemma esetében tették, ami betudható annak, hogy kevesebb a pontkülönbség az egyes stratégiák között, így nem akkora a kockázat. Az is látszik azonban, hogy az leszámítva, hogy nem



4.14. ábra. Nemek Harca stratégiák eloszlása

ismerik fel az ellenfeleik stratégiáját, a GPT-4 és a Gemini az optimális kevert stratégiát játssza ebben az esetben, ami figyelemre méltó. Ezek alapján azt lehet leszűrni, hogy a kitalált játékokat, amiket nem nagyon ismernek, nem nagyon tudják az általuk ismert módszerekkel optimális stratégiákat keresni, azonban az ismert játékokat, mint a fogoly dilemma, vagy a nemek harca, jól alkalmaznak ismert stratégiákat.

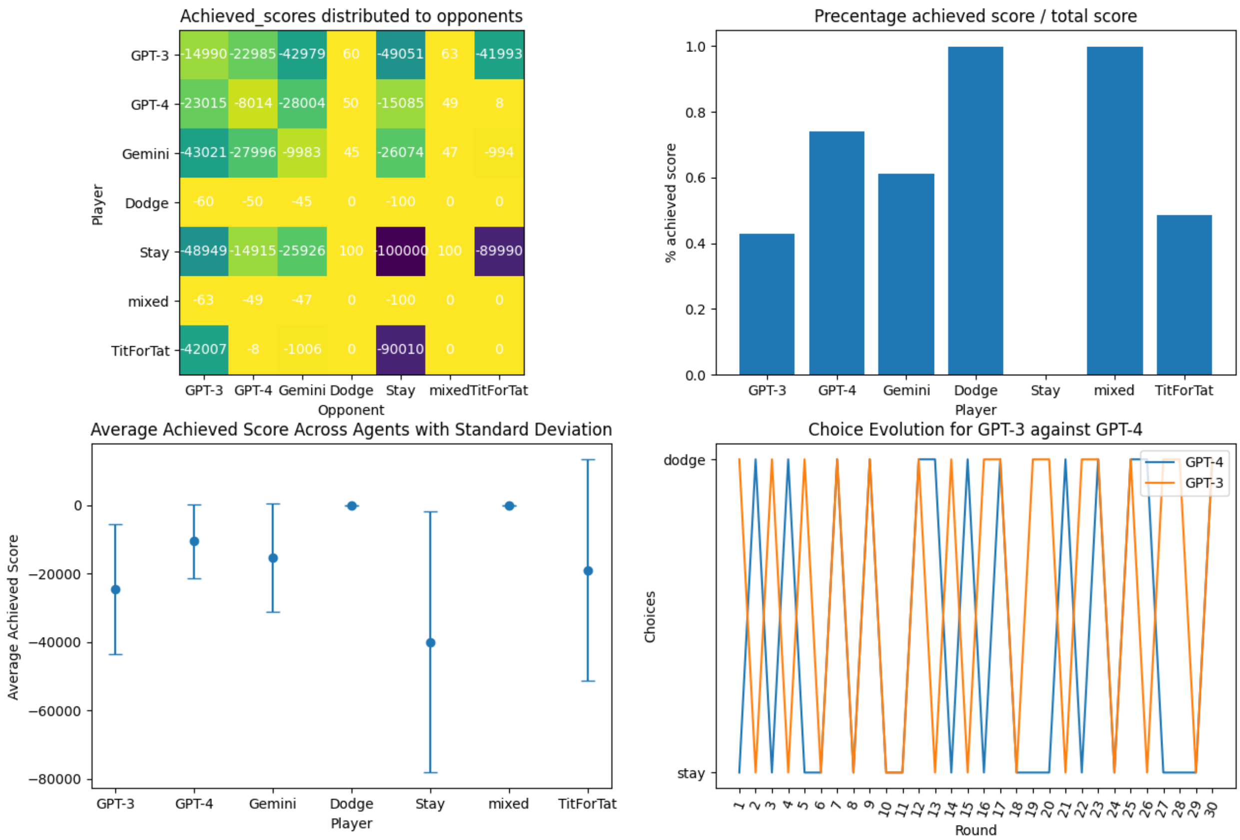
4.2.3. Gyáva nyúl

A gyáva nyúl játékban két autós megy egymással szemben és a játékosoknak azt kell eldönteniük, hogy ki tér ki a másik elöl. Ha valamelyik kitér míg a másik nem, ő lesz a gyáva nyúl, és megvetés tárgya lesz. Amennyiben mind a kettő játékos kitér, egyik játékos sem kap semmi negatív vagy pozitív kifizetést. Azonban, ha egyik játékos sem tér ki, akkor mind a kettő halálos balesetet szenved, tehát nagyon negatív lesz a kifizetésük.

	Kitér	Egyenesen halad
Kitér	$(0, 0)$	$(-1, 1)$
Egyenesen halad	$(1, -1)$	$(-1000, -1000)$

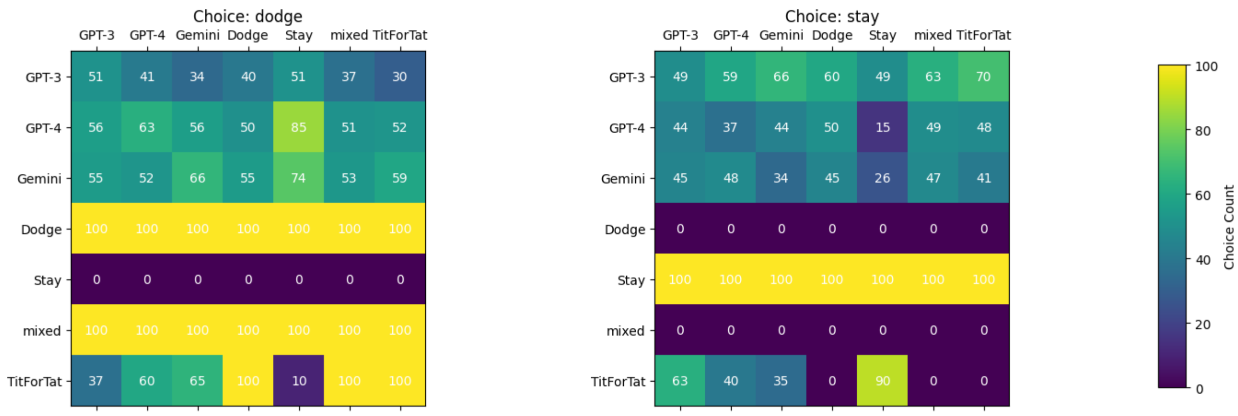
4.15. ábra. Gyáva nyúl játék kifizetési mátrixa

Más névvel is szokták illetni ezt a játékot, illetve az ilyen mátrixú játékokat, mint például a héja-galamb, de az is hasonlónak tekinthető, amikor egy ajtóban két ember találkozik, és azt kell eldönteniük, milyen sorrendben haladnak át. Amelyik elsőbbséget ad, az lassabban jut át, ha mindketten mennek, akkor összeütköznek. A mátrixban található két Nash egyensúly is, méghozzá amikor valamelyik lehúzódik míg a másik nem. Nyilvánvalóan egyik játékosnak sem éri meg módosítani stratégiát hiszen, ha az módosít, amelyik lehúzódik, akkor ütközni fognak, ha pedig az, aki egyenesen halad, csökken a „menőségi faktora”, azaz a kifizetése. Kevert stratégiák közül az értékektől függően elenyésző valószínűséggel éri meg játszani a kockázatos stratégiát, hiszen, ha a másik játékos nem tér ki, végzetes lesz a döntése.



4.16. ábra. Gyáva nyúl eredmények

Nyilvánvalóan az eredményeket tekintve, aki sosem tér ki, általában rossz eredményt ér el, ha van olyan ellenfél, aki akár egyszer is ugyanezt teszi, aki pedig mindig kitér, általában véve nem lesz pozitív pontja, de nem is lesz nagyon negatív. A 4.16. ábrán látszik, hogy ezek a legszélsőségesebb esetek. Egyik esetben szinte szórás nélkül 0 közeli pontszámot ért el a játékos, másik esetben nagy szórással elég negatív pontszámot is el lehet érni, a többi játékos pedig ezek között helyezkedik el. A kevert stratégiát játszó játékos a kicsi valószínűség miatt mindig kitért ezért hasonló eredményeket ért el, mint a folyamatosan kitérő játékos. A nyelvi modellek próbálták váltogatni a stratégiájukat és minél több pontot szerezni, azonban nagyon gyakran ütközés lett belőle és ezért nem végeztek olyan jó helyen. A TitForTat játékos pedig a legtöbb ellenfél ellen jó pontszámot ért el, azonban mivel másolta az ellenfele stratégiáját a sosem kitérő játékos ellen nagyon sok negatív pontszámra tett szert. Ennek ellenére a szórását tekintve ő tudta volna elérni a legmagasabb pontszámot.



4.17. ábra. Gyáva nyúl stratégiák eloszlása

Ebben az esetben a modellek nem remekeltek, mivel nem vették figyelembe azt a nagy negatív pontszámot, amivel az ütközés jár és ahogy az a 4.17. ábrán is látszik, kicsit kevesebb mint a fele alkalommal kockáztattak, emiatt többször is ütköztek, de azért nem voltak olyan rosszak, mint a mindíg maradó játékos.

4.2.4. Példa játék tiszta Nash-egyensúly nélkül

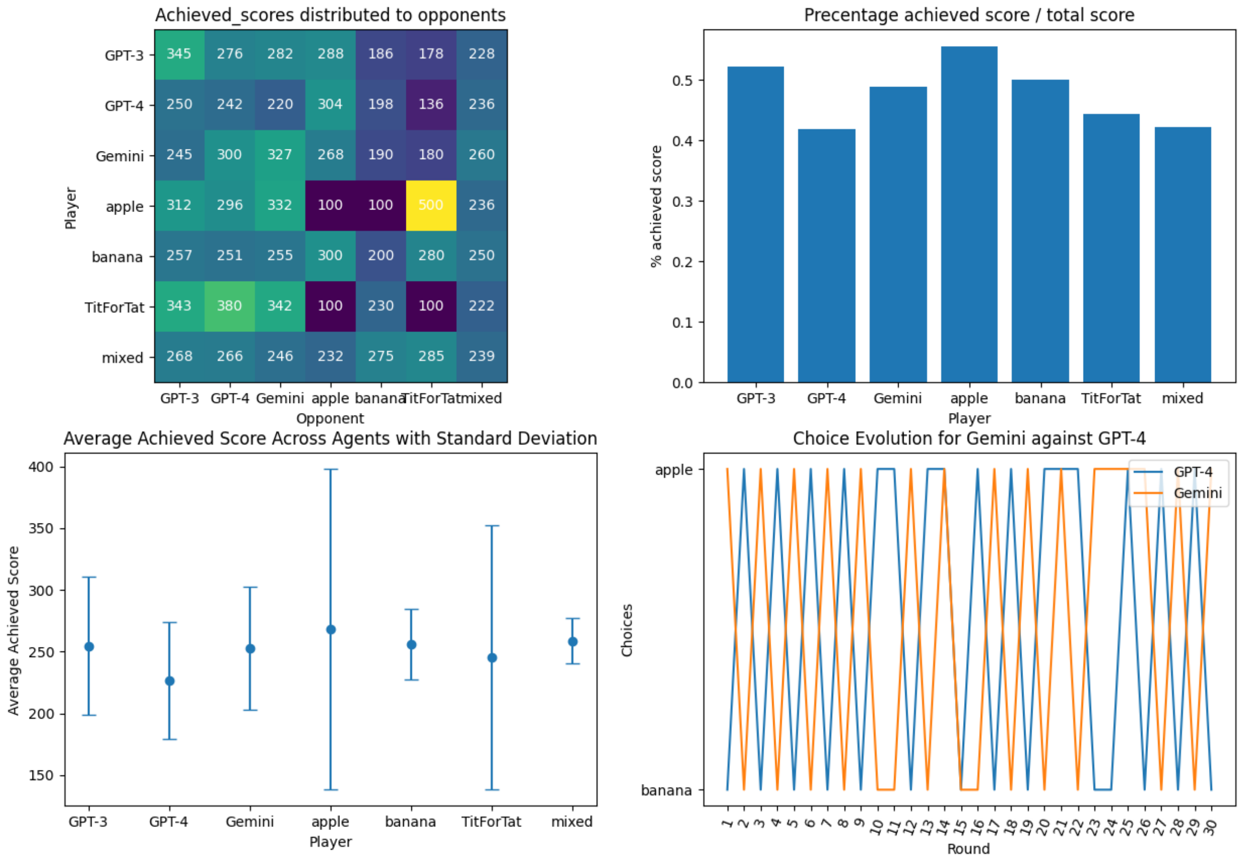
Nem találtam olyan kitüntetett játékot, amely nem zérusösszegű, és nincs benne tiszta Nash egyensúly, de könnyedén lehet ilyen mátrixot kreálni. A következő 4.18. ábrán látható mátrix is egy általam készített játék kísérleti célra.

A tiszta Nash egyensúly hiánya könnyedén ellenőrizhető, minden cellában valamelyik játékosnak megéri stratégiát váltani, a nagyobb kifizetés érdekében. A játékban domináns stratégia sincs, egyedül optimális kevert stratégia van. Eszerint a sorjátékosnak 20% valószínűséggel érdemes a felső sort játszania és 80%-kal az alsót, míg a második, oszlopjátékosnak pedig 33% valószínűséggel az első oszlopot és 66%-kal a második oszlopot.

	Döntés 1	Döntés 2
Döntés 1	$(5, 1)$	$(1, 5)$
Döntés 2	$(1, 3)$	$(3, 2)$

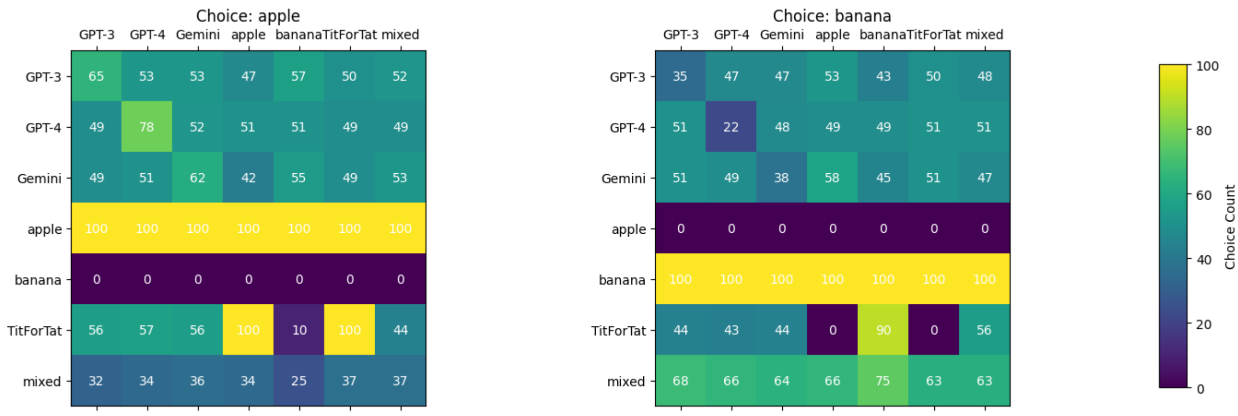
4.18. ábra. Példa nem zérusösszegű játék tiszta Nash egyensúly nélkül.

Ezekkel a stratégiákkal az első játékos várható kifizetése 2.3 lesz míg a második játékosé 2.6.



4.19. ábra. Példa nem zérusösszegű játék Nash egyensúly nélkül eremények

Az átlagokat tekintve (4.19.ábra) jól látszik, hogy valóban 2.5 körül alakul a játékosok pontszámainak átlaga, hiszen az átlagos összesített pontszám 250 körüli. A legkisebb szórással az optimális kevert stratégiát játszó játékos rendelkezik, bár az átlagot tekintve nem az övé a legmagasabb, de az csak amiatt van, hogy a tiszta stratégiát játszó játékosok egymás ellen kiemelkedően sok, illetve kiemelkedően kevés pontot szereznek. Ha stratégiák módosításait tekintjük, úgy tűnhet, hogy a nyelvi modellek próbálnak valamennyire ráilleszteni, illetve pont, hogy eltérni az ellenfelük stratégiájától attól függően, hogy éppen sor vagy oszlopjátékosok azonban, ha egy „egyenes" játékos elleni játékot tekintünk látszik, hogy hiába csak egy stratégiát játszik az ellenfél, a modell nem tudja értelmezni a szituációt és nem játssza a jó stratégiát ellene, tehát ebben a játékban a nyelvi modellek nem teljesítettek jól.



4.20. ábra. Példa nem zérusösszegű játék Nash egyensúly nélkül stratégiák eloszlása

Ha a 4.20. ábrán próbáljuk meg azt vizsgálni, hogy a kevert stratégiát alkalmazzák-e a modellek, akkor csalódnunk kell. Egyrészt még ha azt is játszák nem látszódik jól, mert sorjátékosként más az optimális kevert stratégia, mint oszlopjátékosként, és mivel nagyjából véletlenszerű, hogy melyik játékos melyik, nehezebb visszafejteni, hogy nagyjából jó arányban választottak-e. Másrészről pedig az ábrán az látszik, hogy nagyjából fele-fele arányban választották az egyes stratégiákat a modellek, ami még a kevert stratégiák alapján se jöhet ki. Ez a játék megerősíthet bennünket abban, hogy ismert játékokra képes valamennyire alkalmazkodó stratégiát kreálni a modell, azonban kevésbé neves és kitalált játékokra inkább random választ.

4.3. Kooperatív n személyes játékok

A továbbiakban a mátrixjátékoktól kicsit eltérő játékot próbálunk ki, méghozzá egy kooperációs játékot. A játék lényege, hogy n személy (jelen esetben 3) megy haza taxival. Mindegyik játékos egy irányba lakik és taxival kíván hazamenni. A taxi a költséget távolság alapján számlázza, tehát akár egy embert visz, akár többet, ha út közben kirak egy embert az nem plusz költség, egyszerűen a megtett út árát kell kifizetni. Az egyszerűség/-kiszámíthatóság miatt a költség az egyes játékosokra nézve 6, 12 és 42 dollár, amennyiben külön utaznának, azonban összességében csak a legtávolabbra menő játékos költsége az összköltség amennyiben együtt utaznak, hiszen egy irányba mennek.

4.21. ábra. Vizualizáció a taxi játékhoz

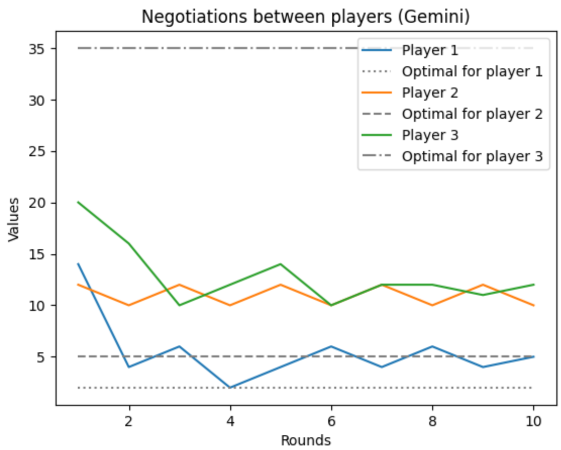
Első lépés ként fontosnak tartottam a különböző nyelvi modellektől megkérdezni, hogy ha ők az 1-es, 2-es, 3-mas játékosok, akkor kivel lépnének koalícióra és miért. Erre olyan válaszokat kaptam legtöbbször a GPT-3 és a Gemini esetében, hogy senkivel sem alkotnának koalíciót, méghozzá eléggé elrugaszkodott, a játékhoz nem kapcsolódó érvekből. Nyilvánvalóan mindenképpen érdemes mind a három játékosnak koalíciót alkotnia ebben az esetben, éppen ezért a következő próbálkozásnál már megszabtam, hogy mindenkivel koalíciót kell alkotnia és szerinte milyen elosztásban lenne igazságos kifizetni a taxiköltséget.

permutációk	1. játékos hozzájárulása:	2. játékos hozzájárulása:	3. játékos hozzájárulása:
1 2 3	6	6	30
1 3 2	6	0	36
2 1 3	0	12	30
2 3 1	0	12	30
3 1 2	0	0	42
3 2 1	0	0	42
átlag:	2	5	35

4.22. ábra. Egyes játékosok hozzájárulása különböző permutációk esetén a taxiköltség játéknál

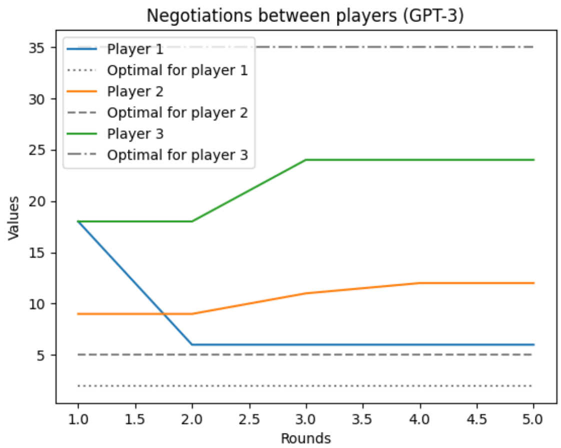
A megválasztott értékek miatt egyébként könnyedén számítható a Shapley-érték a 4.22. ábrán látható módon. A modellek azonban ezt az értéket nem, vagy nem jól tudták kiszámítani. Többszöri próbálkozásra sem az alsó sorban látható értékeket adták ki, sőt voltak olyan esetek is, amikor az értékek még csak nem is 42-re összegződtek. Volt pár olyan próbálkozás, amikor promptok segítségével rávezettem a modellt arra, hogy ez egy kooperációs játék, és ilyen esetben érdemes a Shapley-értéket kiszámítani, azonban hiába tudta a modell a képletet, alkalmazni nem sikerült neki, és nem a jó értékek jöttek ki, még korrekciós prompt után sem.

A végső teszt az volt, hogy a modelleket összeállítottam hármasával tehát például három darab GPT-3, három darab Gemini, stb, illetve kipróbáltam olyat is, hogy mindegyik modellből egyet veszek, és egymással kellett nekik kiegyezni egy végső elosztásban. Míg az eddigieket csak böngészőben kézzel teszteltem, erre külön környezetet hoztam létre, hogy többször is le tudjam szimulálni az egyes modellek alkuképességét. Az információt úgy kapták meg, hogy első alkalommal csak megkérdeztem, hogy mennyit adnának a közös költségbe, ha egyedül önmaguknak csak x összeg lenne. A további körökben pedig megosztottam mindegyik példánnyal, hogy az előző körben ki mennyit ajánlott be, és mennyivel több vagy kevesebb pénzük van jelenleg összesen, mint amennyi az összesített költség lenne.



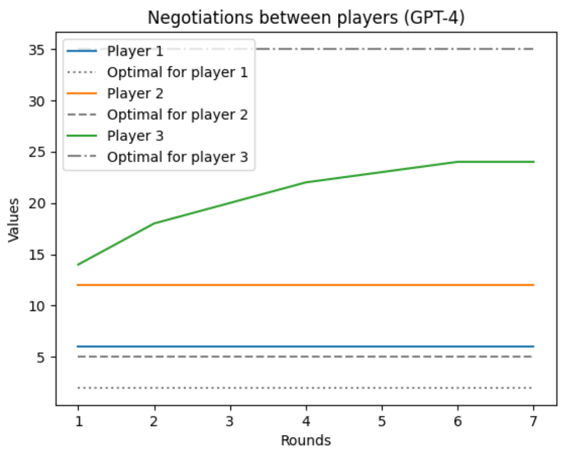
4.23. ábra. Gemini eredmények a taxiköltség játékra

Amikor még nem volt Gemini, hanem Bard volt helyette, a modell nagyon rosszul teljesített ezen a teszten. Gyakran a játékosok még annál magasabb árat is képesek voltak fizetni, mint amennyibe került volna a taxi, ha egyedül utaznak. Ezzel szemben a Gemini már jobb teljesítményt mutat. A 4.23. ábrán látható, hogy a háromból kettő játékos próbál megengedő lenni, de mégsem többet fizetni az alap útiköltségén, azonban a legnagyobb utat megtevő játékos valamiért nem hajlandó engedni felfele, hiába volt nagyobb a kezdő ajánlata, emiatt nem tudnak megegyezni.



4.24. ábra. GPT-3 eredmények a taxiköltség játékra

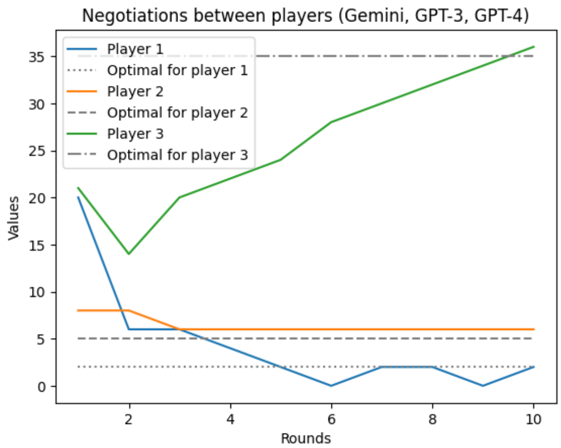
A GPT-3 esetében azt az eredményt szűrhetjük le, hogy a modell kevésbé „makacs", sokkal engedékenyebb, ezáltal hamar, még 5 kör alatt is meg tudott egyezni a három modell, azonban az értékek annál távolabb álltak a Shapley értékektől, sőt a Bard-hoz hasonlóan több pénzt is felajánlottak, mint amennyi az alap költség lett volna.



4.25. ábra. GPT-4 eredmények a taxiköltség játékra

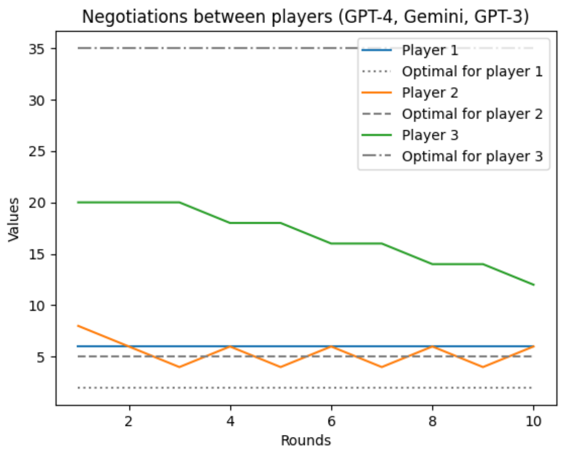
Amikor a három játékos három darab GPT-4 volt érdekes módon két játékos egyáltalán nem módosított az alap költségén, mint ahogy az 4.25. ábrán látszik. Az értékek eloszlása a GPT-3 hoz hasonlóan szintén nem a optimális, hiszen gyakorlatilag csak a harmadik játékos költségeit csökkentették le, a másik két játékos ugyanannyit fizet, azonban ebben az esetben is hamarabb egyeztek meg a játékosok, mint tíz kör.

A legjobb eredmények talán abban az esetben születtek, amikor a három játékos a három különböző modell volt, bár itt is számít a játékosok sorrendje, azaz, hogy melyik



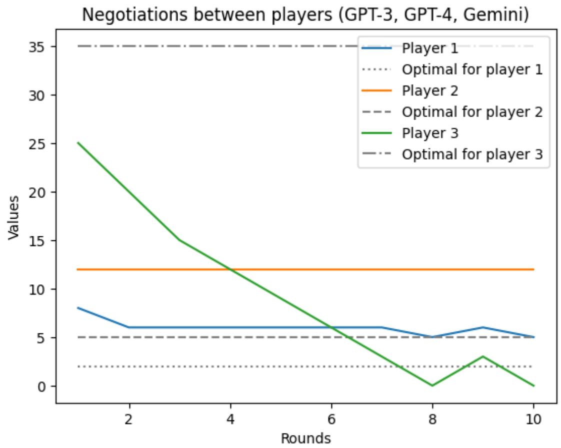
4.26. ábra. Eredmények a taxiköltség játékra rendre Gemini, GPT-3, GPT-4 játékosokkal

modell melyik játékos lesz az adott kísérletben. Az eddigi tapasztalatok alapján, amikor a három különböző modellel próbálkoztam elsőként a Gemini GPT-3, GPT-4 sorrenddel próbálkoztam, hiszen harmadik játékosként a GPT-4 tűnt a legjobbnak a saját környezetében, a Gemini pedig első játékosként volt a legalkuképesebb. Nagyjából jól mértem fel az egyes játékosok „hozzáállását" a játékhoz, mivel ez a sorrend (4.26. ábra) bizonyult a legjobbnak az összes eddigi kísérletből. Habár két dollárral több pénzt gyűjtöttek mint 42, azonban majdnem a Shapley-értékeknek megfelelő számokat ajánlották.



4.27. ábra. Eredmények a taxiköltség játékra rendre GPT-4, Gemini, GPT-3 játékosokkal

A többi permutációt kipróbálva a 4.27. illetve a 4.28 ábrához hasonló eredmények születtek. Több futtatás esetén is a korábbi sorrend jobbnak bizonyult a többinél, ahhoz hasonlóan jó eredményt nem találtam. Feltételezhető ezek alapján a kísérletek alapján, hogy a modellek nem szándékosan találják meg ezeket az értékeket, hanem csak próbál-



4.28. ábra. Eredmények a taxiköltség játékra rendre GPT-3, GPT-4, Gemini játékosokkal

nak megegyezni egy olyan árban, amely egyiküknek sem több mint az alap költség, akár úgy is, ha ez éppen nem igazságos.

5. fejezet

Összegzés

Összességében elmondható, hogy ismételt játékokban a nyelvi modellek nem jól általánosítanak. Amely játékokat már ismerik, azokra képesek jó eredményeket produkálni, azonban újonnan előállított mátrixokra csak random stratégiákat játszanak.

Ami a kooperációs játékokat illeti, elég meglepő, hogy bár a Shapely-értéket ismeri a modell, ami alapján érdemes lenne játszani, használni többféleképpen tesztelve se tudja. Elemezve a kapott eredményeket kicsit olyan érzésünk lehet, mintha mindegyik modell típusnak lenne egy játékstílusa/személyisége. Ez abból is látszik, ahogy permutáltuk az egyes modelleket jól kijött, hogy a megszokott játékstíluson még erős inger ellenére se enged. Ez a jelenség lehet meglepő, nem meglepő, viszont érdemes átgondolni, ha ilyen ingereknek engedne valamelyik modell, az olyan mintha engedne a szociális nyomásnak, ami igencsak egy emberi tulajdonság, tehát azt mutatná, hogy ezek a nyelvi modellek bizonyos helyzetekben emberhez hasonlóan viselkednek.

Ami a jövőbeli terveket illeti, jó lenne több modellen is tesztelni ezeket a kísérleteket, hátha van, ami jobban vagy érdekesebben játszik. Sajnos az infrastruktúra miatt a teljes LLama2 modellt nem sikerült beüzemelni, a kvantált pedig nem működött olyan jól. A későbbiekben, a gépigényeknek eleget tevő számítógéppel meg lehetne kísérelni az LLama-t bevonni ebbe a kísérletbe [1]. A taxis játék esetében is érdekes lehet, ha már nem csak 3 játékos van. Ezen kívül érdemes lehet más felosztásban is lefuttatni ezeket a teszteket, például minden játék csak egy hosszú nem pedig több körös, vagy pont, hogy

[1] Azóta már megjelent a Llama3 modell, melyet az Amazon Claude3 modelljével egyetemben a GPT-4 és Gemini legerősebb konkurenciájának tartanak

hosszabb, hiszen ahogy korábban is említettem, teljesen máshogy kell hozzáállni a játékhoz, ha csak egy körről van szó, vagy többről. Az idő hiányában, illetve a költségek miatt ezeknek a teszteknek a futtatására már nem került sor. A mátrixjátékok esetében általánosságban egy teljes kísérlet futtatása, amelyből a diagramok származnak egyenként nagyjából 45perc futási idővel kell számolni, az összes diagram elkészítése pedig nagyjából 17000 forint az OpenAI modellek API hívásai miatt. A GPT modellek közül is a GPT-4 volt inkább az, amely magas költséggel járt, a GPT-3 hozzá képest elenyésző.

Irodalomjegyzék

[1] Elif Akata, Lion Schulz, Julian Coda-Forno, Seong Joon Oh, Matthias Bethge, and Eric Schulz. Playing repeated games with large language models. *arXiv preprint arXiv:2305.16867*, 2023.

[2] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 2023.

[3] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[4] Caoyun Fan, Jindou Chen, Yaohui Jin, and Hao He. Can large language models serve as rational players in game theory? a systematic analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17960–17967, 2024.

[5] Drew Fudenberg and Jean Tirole. *Game theory*. MIT press, 1991.

[6] Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.

[7] Anna R Karlin and Yuval Peres. *Game theory, alive*, volume 101. American Mathematical Soc., 2017.

[8] Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274, 2023.

[9] Paul R Milgrom. Axelrod's" the evolution of cooperation", 1984.

[10] Guillermo Owen. *Game theory*. Emerald Group Publishing, 2013.

[11] Simon D Parsons, Piotr Gymtrasiewicz, and Michael Wooldridge. *Game theory and decision theory in agent-based systems*, volume 5. Springer Science & Business Media, 2012.

[12] Malik Sallam. The utility of chatgpt as an example of large language models in healthcare education, research and practice: Systematic review on the future perspectives and potential limitations. *MedRxiv*, pages 2023–02, 2023.

[13] Mengwei Xu, Wangsong Yin, Dongqi Cai, Rongjie Yi, Daliang Xu, Qipeng Wang, Bingyang Wu, Yihao Zhao, Chen Yang, Shihe Wang, et al. A survey of resource-efficient llm and multimodal foundation models. *arXiv preprint arXiv:2401.08092*, 2024.

[14] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57, 2024.