

A CHATBOT TECHNOLÓGIA ALKALMAZÁSA MAGYAR NYELVRE

Magnucz Péter László

mérnökinformatikus hallgató, ME-GÉIK Általános Informatikai Tanszék
3515 Miskolc, Miskolc-Egyetemváros, e-mail: peter.magnucz@gmail.com

Baksáné Varga Erika

egyetemi docens, ME-GÉIK Általános Informatikai Tanszék
3515 Miskolc, Miskolc-Egyetemváros, e-mail: vargae@iit.uni-miskolc.hu

Absztrakt

Napjainkban a mesterséges intelligencián alapuló intelligens számítástechnikai rendszerek rohamos fejlődésének lehetünk tanúi. A csevegő robotok, azaz chatbotok, 2011-ben robbantak be a köztudatba, amikor megjelent az Apple Siri virtuális asszisztens, amely az első természetes nyelvet használó felhasználói interfész volt. Azóta számtalan megoldásváltozattal találkozhatunk a webl oldalakon és a mobil eszközök felületein. Előnyük, hogy egyidejűleg korlátlan számú felhasználóval képesek kapcsolatot tartani és a fejlettebbek képesek tanulni is. Léteznek egy adott témakörre specializált csevegő robotok, de vannak általános, humanoid jellegű alkalmazások is amelyek akár több adatbázisból is képesek információt kinyerni és felhasználni. Az eleinte angol nyelvterületen elterjedt chatbotok ma már magyarul is elérhetők. Jelen projekt célja a chatbotok történetének áttekintése és magyar nyelvű megvalósításuk vizsgálata egy mintarendszeren keresztül.

Kulcsszavak: mesterséges intelligencia, csevegő robot, intelligens asszisztens, természetes nyelvek feldolgozása

Abstract

Nowadays we are witnessing the rise of chatbots, which are intelligent software products that use artificial intelligence. Their era started with the emergence of Apple's Siri virtual assistant, which was the first application with a user interface incorporating natural language processing. Since then there have been a tremendous amount of chatbot apps built on websites and in mobile applications. Their main benefit is that they can answer questions from unlimited number of users in real-time and some of them can even learn automatically. There exist domain-specific chatbots and also more general, human-like applications that can infer and apply information from several databases. The first chatbots communicated in English, but by now we can find Hungarian-speaking chatting robots as well. This paper aims at reviewing the history of chatting robots and reports on a test system that was developed for examining their implementation for Hungarian language.

Keywords: artificial intelligence, chatbot, virtual assistant, natural language processing

1. Bevezetés

A chatbot (chatting robot) egy olyan számítógépes program, amely írásban vagy szóban beszélgetést szimulál a felhasználókkal valamilyen üzenetküldő rendszeren – weblapon, mobil alkalmazáson, vagy

telefonon – keresztül [1]. A csevegő robotokra ma használt chatbot kifejezés Michael Mauldintól származik; ő készítette az első szóbeli kommunikációra képes robotot, Juliát 1994-ben [2]. A chatbottal szemben támasztott legfontosabb elvárás, hogy az emberhez hasonlóan képes legyen valós időben interaktív módon párbeszédet folytatni, azaz a felhasználó ne tudja meghatározni, hogy emberrel vagy chatbottal beszélget-e. A chatbotok ezért az alábbi alapvető jellemzőkkel bírnak [3]:

- Párbeszéd-készség: A chatbotnak meg kell értenie a felhasználó kérdését. Ehhez rendelkeznie kell természetes nyelvi feldolgozó (Natural Language Processing, NLP) modullal, ami értelmezi és elemzi a kapott információkat.
- Racionalitás: A chatbot értelmes választ kell adjon a felhasználó kérdésére. Ehhez tárolnia kell olyan kontextus-függő információkat mint például a felhasználó neve, valamint hozzá kell férnie külső tudásbázisokhoz.
- Személyiség: A felhasználó bizalmának elnyerése érdekében fontos, hogy a robot egy valódi személy érzetét keltse. Ezért kapnak a chatbotok nevet és arcot.

A legegyszerűbb chatbotok előre megírt forgatókönyv alapján működnek. Az ilyen programokban a kérdések és az azokra adható lehetséges válaszok előre meg vannak írva, így a "beszélgetések" egy jól meghatározott útvonalon zajlanak. Így működnek a telefonos ügyfélszolgálatok, vagy például a Wish online áruház reklamációkezelő rendszere.

Vannak azonban intelligens chatbotok is, amelyek a kommunikációs stratégiájuk alapján két csoportba oszthatók. Egyszerűbb az implementációja azoknak a robotoknak, amelyek passzív információ szolgáltatók. Ezek előre definiált tudással rendelkeznek egy adott témában, és a válaszaikat ebből az információhalmazból állítják elő. Ilyen a legtöbb ügyfélszolgálati asszisztens, mint például az IKEA Annája.

A másik csoportba az interaktív robotok tartoznak, amelyek képesek akár irányítani is a beszélgetést: visszakérdeznek, és igyekeznek fenntartani a kommunikációt a partnerrel, illetve képesek tanulni és a tanult dolgokat felhasználni a következő beszélgetésben (mint például a Microsoft által fejlesztett Tay és Zo Twitter chatbotok). Igaz ugyan, hogy ezek a rendszerek sem tudják kezelni a kétértelműséget, nem tudnak (vagy csak nagyon lassan) váltani a különböző kontextusok között és a bonyolultabb beszélgetések is gondot okozhatnak nekik, de a velük való beszélgetés egyre jobban hasonlít a személyes párbeszédhez.

2. Történeti áttekintés

A természetes párbeszédet szimuláló intelligens botok története 1950-re nyúlik vissza, amikor Alan Turing megalkotta azt a tesztet, amivel igazolható, hogy egy számítógép program képes-e intelligens viselkedésre [4]. A teszt lényege, hogy egy zárt szobában lévő ember, aki távolról kommunikál egy számítógéppel és egy valós személlyel, mennyi idő után jön rá, hogy melyik beszélgető partnere a gép és melyik az ember.

1966-ban Joseph Weizenbaum létrehozta az ELIZA névre keresztelt beszélgető robotját [5]. Bár a Turing-teszten elbukott, a beszélővel el tudta hitetni magáról, hogy ember. Az ELIZA jelentősége abban áll, hogy a rendszer kialakítása során alkalmazott elvek megalapozták a chat robotok későbbi struktúráját, mint például a kulcsszavak és a konkrét kifejezések felismerése, valamint az előre programozott válaszok.

1972-ben Kenneth Colby bemutatta PARRY-t, ami egy skizofrén személyt szimuláló program volt [6]. A tesztelésében résztvevő pszichiáterek 48%-a tudott csak különbséget tenni PARRY és egy valódi személy között.

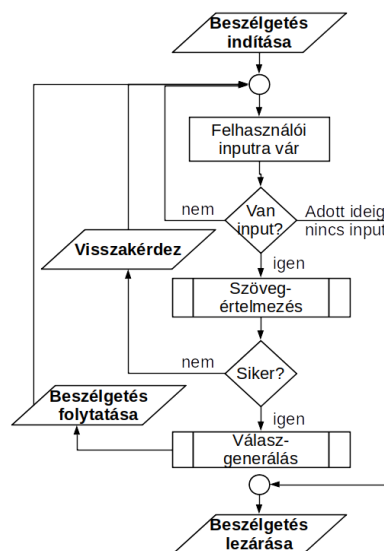
1995-ben megjelent A.L.I.C.E. (Artificial Linguistic Internet Computer Entity), az első beszélgető robot, ami nyelvi feldolgozót alkalmazott [7]. Válaszait a tudásbázisában tárolt < minta > (input) – < válasz > (output) párok alapján állította össze. A mintákat tartalmazó dokumentumokat AIML (Artificial Intelligence Markup Language) nyelven [8] írták, ami egy XML kiterjesztés és mai napig használják egyszerűbb chatbotok írásához. Annak ellenére, hogy a Turing-teszten elbukott, számos elismerést kapott, mert ez volt abban az időszakban a legfejlettebb chatbot.

2001-ben született a Smarterchild chatbot, ami sok tekintetben az Apple Siri és a Samsung S Voice előfutárának tekinthető. A következő évtizedben a chatbotok jelentős fejlődésnek indultak és nagyon népszerűek lettek a vezető IT vállalatok körében: Apple Siri (2011), Google Now (2012), Amazon Alexa (2015), Microsoft Cortana (2015). Ezek a botok többek között képesek reagálni a hangutasításokra, zenét lejátszani és internetes kereséseket végezni. A kapott szövegben nem csupán kulcsszavak alapján keresnek, hanem neurális hálók és gépi tanulási módszerek alkalmazásával elemzik azt. A 2016-ban megjelent Google Assistant már kétirányú kommunikációra is képes.

A magyar nyelvterületen egyedülálló az általános beszélgetést imitáló Lafoxka, ami a mindennapi beszéd írott formáját felismerve érdekes és fordulatos válaszokat produkál [9]. A program fix tudásbázissal rendelkezik, amelyet a szerző (Schmidt Ágnes) több mint 5 éve folyamatosan bővít, a kilencszázezer sornál több begépelte beszélgetés alapján, amelyet Lafoxkával folytatott a több mint kilencezer bejelentkező, tizenhétvezernél több beszélgetés során [10]. A program működési elve alapján felismeri a jellegzetes szófordulatokat a beszélgető partner szövegében, majd kikeresi tudástárából az ezekre adható válaszok közül a kontextushoz legjobban illeszkedőt. Ennek érdekében a program nemcsak az utolsó mondatot elemzi, hanem a beszélgetés során elhangzott témákat is rögzíti.

3. Virtuális asszisztens megvalósítása

Napjainkban a legtöbb online megjelenéssel rendelkező cég alkalmaz virtuális asszisztens robotot. A magyar nyelvű webes és mobil felületeken természetesen magyarul szólítják meg a felhasználókat. Egy tipikus beszélgetés menetét az 1. ábra szemlélteti, mely során az 1. táblázatban összefoglalt nyelvi minták fordulnak elő.



1. ábra. Virtuális asszisztens általános működési modellje.

A szövegértelmező modul feladata a felhasználó által megadott szöveg gépi elemzése: az abban rejlő információk automatikus kigyűjtése, rendszerezése és szűrése. Egy információkinyerő rendszer (Information Retrieval, IR) nyelvfüggő és három alapvető komponens alkotja: az adatbázis, a kereső motor és a felhasználói felület [11]. A virtuális asszisztensek működésére a zárt-világ modell jellemző, azaz egy jól behatárolt területen alkalmazhatók. Erre a területre vonatkozó információkat tárolnak az adatbázisukban, és csak ritkán tartalmazznak gépi tanuló algoritmust. Az adatbázis szerkezete kulcsfontosságú a hatékony visszakereséshez és rendszerint (input minta, válasz minta) párokat tartalmaz. A kereső motor szavakra bontja az input szöveget és például a magyar nyelv esetén szótövezést is végez (eltávolítja a szavak végéről a toldalékot, azaz megkeresi a szóalakok szótövét), majd kikeresi az adatbázisból az input mintára illeszkedő válaszokat.

A szövegértelmező által visszaadott válaszlehetőségekből a válaszgeneráló modul állítja elő a felhasználó számára megjelenítendő információt. A keresés típusától függően ez lehet az összes találat megjelenítése, vagy egy kontextus-alapú választás a találati listából [12].

1. táblázat. Példák magyar nyelvű virtuális asszisztens szóhasználatára

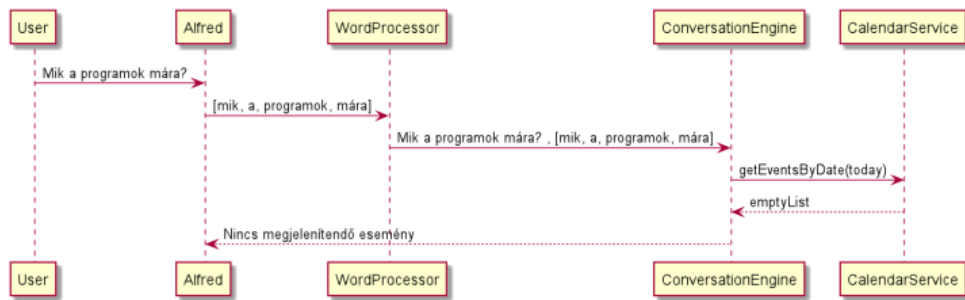
Kontextus	Szöveg minta
Beszélgetés indítása	Szia! Miben segíthetek?
Visszakérdezés: a felhasználó által megadott szövegben nem talált az adatbázisában szereplő input mintát. A kérdés átfogalmazását, a téma pontosítását várja. A kérdésben gyakran megismétli a felhasználó által használt, számára ismeretlen fogalmat.	Milyen ... gondolsz?
Beszélgetés folytatása	Tudok még valamiben segíteni?
Beszélgetés lezárása	Ha nincs több kérdésed, elbúcsúzom. Viszlát!

4. Alfred – A magyar személyi asszisztens

Célunk egy olyan személyi asszisztens chat robot létrehozása, amely segít beosztani és ütemezni a feladatainkat a Google naptár felhasználásával. Alfred nem alkalmazásként lett implementálva, azaz nem rendelkezik önálló felhasználói felülettel, hanem szolgáltatásként fut. Vele kapcsolatba lépni, funkcióit igénybe venni a Slack [13] nevű alkalmazás segítségével lehet. A beépülő alkalmazás létrehozása után regisztráltunk egy „bot user”-t, amihez a Simple Slack API-n [14] keresztül lehet csatlakozni: üzeneteket küldeni és fogadni.

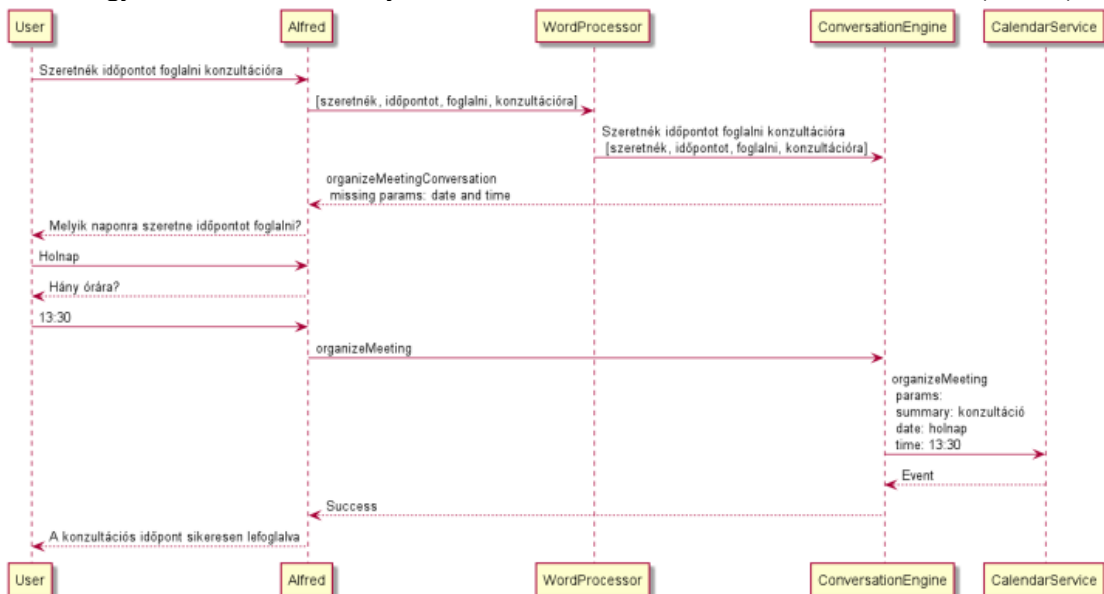
4.1. Funkciói

A chatbot képes a felhasználóhoz tartozó naptárat olvasni, azaz lekérdezni a benne szereplő adatokat. Az események lekérdezése történhet a mai napra, egy megadott tetszőleges napra (például az alábbi kifejezések használatával: „holnap”, „jövő kedd”), vagy dátum alapján. Ha van olyan esemény, amely a keresési kritériumnak megfelel, akkor tudatja a felhasználóval az esemény nevét és időpontját. Amennyiben nincs a keresésnek megfelelő eredmény, akkor erről értesíti a felhasználót szöveges üzenet formájában (1. ábra).



2. ábra. Egy adott nap programjainak lekérdezése.

A chat robotnak szerkesztenie is kell a naptárat, amikor új eseményt szeretne rögzíteni a felhasználó. Ebben az esetben Alfred bekér minden kötelező paramétert, amely szükséges az esemény létrehozásához, illetve lehetőség van az opcionális paraméterek megadására is. Új esemény létrehozásakor kötelező paraméterek: az esemény neve, kezdő és záró időpontja. Opcionálisan megadható paraméterek: a résztvevők listája, a szabad szöveges leírás és az esemény helyszíne. Sikeres hozzáadás esetén megjelenik az esemény a naptárban, és erről küld a chatbot visszajelzést e-mailben. Amennyiben hiba lépne fel, vagy nem lehet az eseményt létrehozni a robot erről is értesíti a felhasználót (2. ábra).



3. ábra. Új esemény létrehozása.

Az értekezletek és találkozók megszervezése általában hosszú és bonyolult folyamat, hiszen több emberrel kell egyeztetni amikor egy mindenki számára megfelelő időpontot kell találni. Ebben nyújt segítséget Alfred találkozásszervező képessége. Egy találkozó megszervezéséhez meg kell adni a meghívni kívánt személyek e-mail címét, egy helyszínt és egy tervezett időpontot. Ezek alapján Alfred előállít egy meghívót, amit e-mailben elküld a címzetteknek. Az érintettektől a meghívás jóváhagyását vagy visszautasítását várja, és feldolgozza a visszaérkező válaszokat. Amennyiben mindenki jóváhagyta az időpontot, a naptárban rögzíti az eseményt. Ha viszont valaki visszautasítja a meghívót, akkor egy új (a találkozót kezdeményező felhasználó naptárában szabad) időponttal új meghívót készít és

újrászervezi a találkozót. Ezt addig folytatja, amíg nem sikerül rögzíteni egy mindenki által jóváhagyott időpontot a naptárban.

Alfred konzultációkat is képes szervezni automatikusan. Ezek olyan események, amelyekhez nem kapcsolódnak konkrét meghívottak, a robot fogja beosztani a jelentkezőket. A felhasználónak meg kell adnia a konzultációs időpontját és erre az időre várja a chatbot a jelentkezéseket e-mailben. Amikor beérkezik egy igény, Alfred besorolja a kérést és megbecsülni mekkora időablakra van szükség a konzultációhoz. Ezután a kérések beérkezésének sorrendjében foglalja az időpontokat a konzultációra, amelyeket beiktat a felhasználó naptárába. Amennyiben egy jelentkező mégsem tud a konzultáción részt venni, elég Alfrednak üzennie és a program automatikusan kitörli a foglalást, majd egy következő konzultációs időpontot ajánl fel.

Chatbotunk negyedik funkciója a naptárban szereplő események automatikus átütemezése. Amikor váratlan esemény miatt egy adott nap programjait le kell mondjuk és másik napra kell áthelyezzük, Alfred elvégzi helyettünk ezt a szervezési feladatot. Az adott napra eső minden eseményt automatikusan áthelyez a következő szabad helyre a naptárban. Ha egy adott esemény találkozói vagy konzultáció típusú, vagyis más személyeket is érint a program elhalasztása, Alfred értesítéseket küld e-mailben és találkozó esetén újraegyeztet az új időpont jóváhagyása érdekében.

4.2. A szövegfeldolgozó modul működése

A legegyszerűbb megoldás, ha a felhasználó által megadott szövegben teljes egyezőségeket keresünk. Ilyenkor számít a szavak toldalékolása, a kis- és nagybetű használat és nem alkalmazunk szinoníma szótárat. Amilyen egyszerű azonban a megvalósítása, olyan körülményes egy ilyen rendszer kezelése a felhasználó szemszögéből nézve. Ezért mindenképpen célszerű természetes nyelvi feldolgozó alkalmazása. Az elérhető magyar nyelvű NLP rutinokat Orosz György gyűjtötte össze [13], mi azonban saját szövegfeldolgozó modul készítése mellett döntöttünk.

Ennek az egységnek a feladata, hogy felismerje a szövegben a számokat, a dátumokat, tisztában legyen a napokkal, kezelje a szinonimákat, illetve elvárjuk, hogy a felhasználó elgépelési hibáival szemben valamennyire toleráns legyen. A beérkező üzenet feldolgozása a következő lépésekben történik meg:

1. A szöveg egységes formára hozása.
2. Darabolás szavakra.
3. Beszélgetés témájának meghatározása a szavak és a teljes szöveg ismeretében.
4. Válasz reakció elindítása.

A felhasználó által érkezett üzenetet első lépésben csupa kisbetűssé alakítja. Erre azért van szükség, mert a kódszótárban csupa kisbetűvel szerepelnek a szavak, amiket ismer a program. Ezt követően a szöveget szavakra bontja.

A szavak feldolgozása során először a dátumokat szűri ki a szövegből és tárolja el, mert ezek lesznek a végrehajtani kívánt utasítás paraméterei. Amennyiben nem szerepelt dátum a szövegben, akkor továbblép és a beszélgetés témájának beazonosításával folytatódik a feldolgozás. A beszélgetés témájának azonosítása a felbontott szavak és a teljes szöveg alapján történik. A betanult és eltárolt témakörhöz tartozó kulcsszavakat hasonlítja a paraméterben érkezökhöz, majd a legpontosabb egyezéshez tartozó témakör kerül kiválasztásra.

Az egyezés vizsgálatához a Damerau-Levenshtein távolságszámítást használtuk [16]. Ez a szavak közötti szerkesztési távolságmérték a hagyományos Levenshtein-féle szerkesztési távolság egy változata. Annyiban különbözik az elődjétől, hogy egy további művelet is megengedett, mégpedig a karaktercsere. Ez nem a hagyományos csere, ahol egy karaktert kicserélhetünk egy bármilyen másikkra

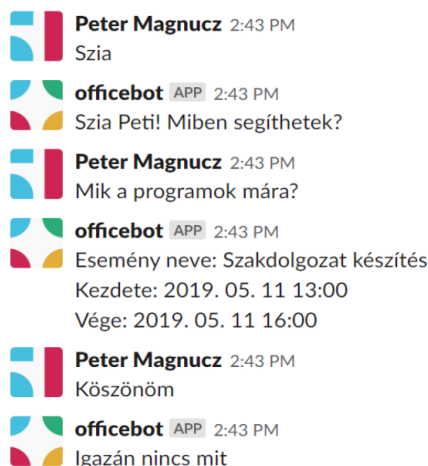
1 költségen, hogy közelebb kerüljünk a másik sztringhez, hanem két szomszédos karakter felcserélése az egyik sztringben ugyanilyen költségen. Damerau állítása szerint ennek a négy szerkesztési műveletnek (a Levenshtein-féle karakter törlés, beszúrás és csere, valamint a szomszédos karakterek felcserélése) az együttes alkalmazásával lefedhető az emberi elgépelésből származó hibák 80%-a [17].

Amennyiben egyik mintára sem illeszkedik a kapott szöveg, akkor egy alapértelmezett válasz akció megy végbe, mely során a csevegő robot pontosítást kér, mert nem tudta feldolgozni a kapott szöveget. Amennyiben sikeresen meghatározta a beszélgetés témáját, akkor végrehajtódik a válasz akció, amely lehet információ kérés vagy a kívánt utasítás végrehajtása. Ha az utasításhoz minden paraméter rendelkezésre áll, akkor az utasítás végbemegy. Abban az esetben viszont ha hiányzik valamelyik paraméter, például a dátum, akkor a chatbot rákérdez erre, majd az eredményt eltárolja és végrehajtja az utasítást.

4.3. Teszt beszélgetés Alfreddel

Most bemutatjuk, hogy működik egy adott nap programjainak lekérdezése. A beszélgetés úgy indul, hogy a felhasználó megszólítja a személyi asszisztenst. Erre Alfred válaszol és felteszi a kérdést, hogy miben segíthet. Várja a felhasználó kérdését. Az „üzenet érkezett” eseményt folyamatosan figyeli és elkapja, majd megvizsgálja, hogy a feladó nem-e önmaga. Ebben az esetben nem folytatódik a feldolgozás, mivel a bot saját üzeneteivel nem foglalkozik.

Legyen a felhasználó kérdése most: „Mik a mai programok?”. A program fogja a kapott szöveget, és feldarabolja szavakra, azaz átalakítja szavak listájává amelynek elemeit azután csupa kisbetűssé konvertálja. Ezek után a *processWord()* metódushoz kerül a vezérlés, mely meghatározza a dátumokat a szövegben. Mivel szerepel a szavak között a "mai" kifejezés, ezért az input paraméterekbe bekerül a mai dátum.



4. ábra. Mai programok lekérdezése a Google naptárból.

Ezt követően a vezérlés átkerül a *conversationEngine*-hez, ami elsőként felépíti a beszélgetés keretét: ez a pozitív választ és a pontosító kérdést tartalmazza. Mivel jelen esetben rendelkezésre áll minden paraméter, így nem kérdez vissza. Ezután a beszélgetés témakörének beazonosításával folytatja a feldolgozást. A *conversationEngine* megvizsgálja a kapott szavak alapján, hogy melyik eltárolt be-

szélgetés témához tartozó kulcsszavakkal egyezik leginkább a kapott szólista és ennek a beszélgetésnek hívja meg az *answer()* metódusát.

Itt a kontextus paraméterei közül kivesszi a felhasználó által megadott dátumot, majd az időpontot beállítja 0:00-ra, mert az időpontra vonatkozóan nem adott a felhasználó megkötetést, azaz alapértelmezés szerint a lekérdezés az egész napra vonatkozik. Ezt követően a *calendarService* segítségével megszólítja a Google Calendar API-t [18], hogy az adott napra eltárolt eseményeket szeretné lekérdezni. Amennyiben nincs erre a napra rögzített esemény a naptárban, „Nincs még semmilyen esemény rögzítve” üzenettel válaszol a bot. Ellenkező esetben visszaadja az esemény(ek) címét és kezdő valamint záró dátumát; és ha nincs további kérése a felhasználónak, akkor elköszön.

5. Összefoglalás

A csevegő robotok készítéséhez szükséges technológiai háttér a Facebook Messenger alkalmazás 2011-es megjelenésével vált mindenki számára elérhetővé. Ettől kezdve rohamos fejlődésnek indult ez az ágazat és a chatbotok elárasztották a webet és a mobil alkalmazásokat. Elsődleges felhasználási területük az információszerzés automatizálása. Cikkünk egy olyan személyi asszisztens megvalósításának részleteit ismerteti, amely a Google Calendar naptárszolgáltatással összekapcsolva képes csoportos találkozókat és személyes konzultációkat szervezni, illetve a naptárban rögzített eseményeket lekérdezni a felhasználó kérésére. Felhasználói felülete magyar nyelvű, ami a kisebb elgépelési hibákra nem érzékeny és az eltárolt szinoníma listának köszönhetően nemcsak az előre betanult kérdésekre képes válaszolni. Természetesen a tökéletesen működő, minden természetes nyelvi szófordulatot ismerő chatbot megalkotásától még távol vagyunk, de az elkészült teszt rendszer racionális kommunikációs környezetben megbízhatóan elvégzi a definiált feladatkörét. Tovább-fejlesztési célunk a chatbothoz külső szótár készítése ami lehetőséget biztosít a többnyelvűsítésre.

6. Köszönetnyilvánítás

A cikkben ismertetett kutató munka az EFOP-3.6.1-16-2016-00011 jelű „Fiatalodó és Megújuló Egyetem – Innovatív Tudásváros – a Miskolci Egyetem intelligens szakosodást szolgáló intézményi fejlesztése” projekt részeként – a Széchenyi 2020 keretében – az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Irodalom

- [1] Tusnádi, I.: Mi az a Chatbot és érdemes használni? <https://www.usernet.hu/blog/mi-az-a-chatbot-es-erdemes-hasznalni> Letöltve: 2020.07.05.
- [2] Mauldin, M.L.: ChatterBots, TinyMuds, and the Turing test: entering the Loebner Prize competition, in AAI '94: Proceedings of the twelfth national conference on Artificial intelligence (vol. 1), pp. 16–21 (1994)
- [3] Sansonnet, J.-P., Leray, D., Martin, J.-C.: Architecture of a Framework for Generic Assisting Conversational Agents, Intelligent Virtual Agents. Lecture Notes in Computer Science 2006: 145-156. http://doi.org/10.1007/11821830_12
- [4] Turing, A.: Computing Machinery and Intelligence. *Mind* 1950, 59(236):433-460. <http://doi.org/10.1093/mind/LIX.236.433>
- [5] Weizenbaum, J.: Computer Power and Human Reason: From Judgment to Calculation. New York: W. H. Freeman and Company. ISBN 0-7167-0464-1 (1976).

- [6] Colby, K.M., Hilf, F.D., Weber, S., Kraemer, H.: Experimental validation of a computer simulation of paranoid processes. *Mathematical Biosciences* 1972, 15(1-2):187–191. [http://doi.org/10.1016/0025-5564\(72\)90073-9](http://doi.org/10.1016/0025-5564(72)90073-9)
- [7] Wallace, R.S.: The Anatomy of A.L.I.C.E., In Epstein, R.; Roberts, G.; Beber, G. (eds.) *Parsing the Turing test*. London: Springer Science+Business Media. pp.181-210 (2009). http://doi.org/10.1007/978-1-4020-6710-5_13
- [8] AIML Docs (AIML Foundation), <http://www.aiml.foundation/doc.html> Letöltve: 2020.07.25.
- [9] VISSZABESZÉL - Édességpárti társalgószoftver, Transindex.ro <http://www.lafoxka.hu/media/transindex.htm> Letöltve: 2020.07.28.
- [10] Lafoxka, <http://www.lafoxka.hu/> Letöltve: 2020.07.28.
- [11] Chu, H.: *Information representation and retrieval in the digital age*, Medford, NJ: Information Today for the American Society for Information Science and Technology (2003). ISBN 1573871729
- [12] Dodiya, T.; Jain, S.: Comparison of Question Answering Systems, In: Abraham A., Thampi S. (eds) *Intelligent Informatics, Advances in Intelligent Systems and Computing*, vol 182. Springer, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-32063-7_12
- [13] Slack Technologies, <https://slack.com> Letöltve: 2019.09.08.
- [14] Ulink, Simple slack API, <https://github.com/Ulink/simple-slack-api> Letöltve: 2019.09.08.
- [15] Orosz, Gy.: *Awesome NLP Resources for Hungarian*, <https://github.com/oroszy/awesome-hungarian-nlp> Letöltve: 2020.07.27.
- [16] Brill, E.; Moore, R. C.: An Improved Error Model for Noisy Channel Spelling Correction, *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pp.286–293 (2000). <http://doi.org/10.3115/1075218.1075255>
- [17] Damerau, F. J.: A technique for computer detection and correction of spelling errors, *Communications of the ACM*, **7** (3): 171–176 (1964). <http://doi.org/10.1145/363958.363994>
- [18] Google Calendar API, <https://developers.google.com/calendar/v3/reference/> Letöltve: 2019.09.08.