Homoki Péter[1]

Miként lehet a szöveggeneráló eszközöket a jogászi hivatások körében hasznosítani?

Absztrakt: A tanulmány a magyar jogászok szempontból kívánja megvizsgálni, hogy a szöveggeneráló eszközöket miként lehetne a jogi munkában hasznosítani. Ennek részeként bemutatja ezen eszközök főbb típusait, felosztásait (mind a szakirodalom, mind a piaci termékek szempontjából), valamint kísérletet tesz annak megmutatására, hogy mi az az irány, ahol a jogászok számára is jelentős fejlődést lehetne elérni, és melyek lehetnek a fejlődés, az ilyen eszközök szélesebb körű használatának korlátai a magyarhoz hasonló méretű nyelvi és jogi közegben.

Tárgyszavak: bérleti szerződés; BERT; ClauseBase; ContractExpress; diskurzus alapú szöveggeneráló eszköz; dokumentum összeállító szoftver; eljárási alapú eszköz; GPT-3; HotDocs; natural language generation; sablon; sablon alapú szöveggeneráló eszköz; szemantikai jelölőnyelv; szöveggeneráló eszköz; szövegkivonatoló eszköz; Woodpecker; XpressDox.

Bevezetés

Ebben a fejezetben a szöveggeneráló eszközök jogászi munkában történő használatát, ennek lehetőségeit, korlátait és feltételeit fogom elemezni. A szöveggeneráló eszközök nagy múltra tekintenek vissza, és a jog területén is már az 1960-as évektől használják őket. A pontos fogalmat és fajtákat később elemzem, munkafogalomként csak annyit bocsátok előre, hogy a szöveggeneráló eszközök olyan szoftverek, amelyek általában strukturált dokumentumokat, szövegeket képesek automatikusan vagy félautomatikusan létrehozni.

[1] Homoki Péter ügyvéd, szabályozási szakjogász. Az Eötvös Loránd Tudományegyetem Állam- és Jogtudományi Karának elvégzése óta kizárólag informatikai és távközlési jogi ügyekben dolgozik, 2008-tól szabályozási szakjogász képesítéssel. Az Európai Ügyvédi Kamarák Tanácsa (CCBE) informatikai jogi szakosztályának elnöke 2012–2018 között, 2019-ig pedig a Magyar Ügyvédi Kamara informatikai megbízottja volt. A CCBE szakértőjeként dolgozik informatikai projektekben, így az AI4Lawyers felmérésben. Ügyvédként informatikai tárgyú beszerzési, kiszervezési és szabályozási témakörökben jártas, különösen a pénzügyi iparágban, valamint kiemelt figyelemmel követi az eIDAS és e-ügyintézési szolgáltatásokkal kapcsolatos tárgykört. Informatikai fejlesztői tapasztalattal rendelkezik a legaltech területen, elsősorban dokumentumautomatizálási témakörökben.

https://doi.org/10.36250/01071_07

Ezeket a rendszereket a számítástechnika azon területére szokták besorolni, amelyet „természetes nyelvi szövegalkotásnak" (*natural language generation* – NLG) neveznek. Az NLG-t a természetes nyelvi *feldolgozás* egyik alkalmazási területeként szokták megjeleníteni, szembeállítva a szöveg*elemzéssel*.[2] A szembeállítás lényege, hogy ha a gépi *szövegelemzés* célja az, hogy egy természetes nyelvű konkrét szövegből (szövegfelszínből) automatizált következtetéseket vonjanak le (annak bármilyen szintű nyelvtani szerkezetéről, tartalmáról, jelentéséről vagy akár a szöveg alkotójának szándékáról), akkor a *nyelvgenerálás* célja ennek az ellentétje: valamilyen számítógépi adatból (ábrázolásból) minél természetesebbnek ható szöveget vagy emberinek hangzó beszédet hozzanak létre.

Ebben a fejezetben a nyelvgenerálás problémakörén belül inkább a szöveggenerálás problémakörére fogok összpontosítani, hiszen a szövegből beszéd automatikus generálásának kevés jogi specifikuma van (hasonlóan a beszéd gépi szöveggé átalakításához). Lehetne e körben is kutatni, csak a keresztmetszet elég szűk (olyan területek tartoznak ide, mint a törvényszéki retorika, az automatizált szövegfelolvasás stb.).

A hétköznapi felhasználó számára a gépi szövegalkotás legismertebb felhasználási területe talán a gépi fordítás[3] és a különféle csevegőrobotok és virtuális asszisztensek területe.[4] A szövegalkotó eszközök felhasználási köre azonban ennél jóval szélesebb körű. A gépi szövegalkotáshoz tartoznak a tömeges levél- vagy szöveglétrehozó alkalmazások, beleértve a nagy ügyfélszámú vállalatok szerződéses iratainak sablonalapú létrehozását és annak folyamattámogatását is (ezt „*document assembly*", néha pedig „*computer-aided drafting*" néven említik),[5] vagy olyan teljesen más természetű alkalmazások is, mint amikor a beérkező időjárási adatokból természetesnek hangzó szöveges előrejelzést készítenek, esetleg pénzügyi vagy katonai adatokból jelentéseket készítenek stb.[6] Idesorolják a kérdés-felelet típusú rendszereket is, de az ilyen rendszerek esetén a rövid elvárt válaszokra figyelemmel a fókusz inkább a helyes információ fellelésén van – ettől még ez szövegalkotási feladat is egyben.[7] Ugyanígy a szövegalkotáshoz tartoznak egy nagyobb szöveg összefoglalását, a „lényeg" kiemelését célzó megoldások is *(text summarization)*.[8]

[2] Nitin Indurkhya – Fred J. Damerau (szerk.): *Handbook of Natural Language Processing*. Boca Raton, Chapman & Hall/CRC, 2010. 5.

[3] Daniel Jurafsky – James H. Martin: Speech and Language Processing. (H. n.), (k. n.), 2020. 203–230.

[4] Jurafsky–Martin (2020): i. m. 492–525.

[5] Marc Lauritsen: Current Frontiers in Legal Drafting Systems. A Working Paper for the 11th International Conference on AI and Law. Stanford, Stanford University, 2007. 1.

[6] Indurkhya–Damerau (2010): i. m. 530–555.

[7] Jurafsky–Martin (2020): i. m. 464–490.

[8] Jurafsky–Martin (2020): i. m. 197; Yang Zijian Győző – Perlaki Attila – Laki László János: Automatikus összefoglaló generálás magyar nyelvre BERT modellel. XVI. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, 2020. 343.

*A szöveggeneráló megoldások felépítése (architektúrája) jelentős mértékben fü*gg a felhasználási területtől: jelentős eltérések vannak a követelményekben és a gazdaságossági kívánalmakban. Teljesen mások a megalkotott szöveg minőségével szembeni elvárások, ha egy átlagos chatbotról van szó, ami legrosszabb esetben egy rosszul sikerült PR-kísérletté válik, és más, ha egy orvosok számára készült szöveges jelentés lesz a végeredmény, amelyen több ember élete múlik. Ugyanígy más a szövegalkotási automata célszerű felépítése, ha egy néhány száz elemes lehetséges kérdéskört kell megfelelően beazonosítani, majd arra egy pár mondatos választ adni, mint ha az elvárt válasz mérete többoldalnyi szöveg. A gazdaságossági követelmények is határt szabnak az egyes megoldások felhasználhatóságának. Ha még ma is igaz volna, hogy egy szofisztikált NLG-rendszernek éves szinten legalább tízezer oldalt kell generálnia, hogy a gyakorlatban megérje létrehozni és fenntartani,[9] akkor ez egyben súlyos ítélet lenne a tízmilliós „kis" nyelvekre is,[10] és nem csak a jogi doménen belül, hanem bármilyen szakterületen.

Minderre tekintettel talán nem fölösleges az a vállalkozás, hogy gyakorló jogászként önző szakmai nézőpontból kíséreljük meg áttekinteni, mi mindenre érdemes ezeket a technikákat használni. E vizsgálatok során figyelembe kell venni az elérhető piaci megoldásokat, valamint egyes szakirodalmi álláspontokat, és egyben nagyvonalú gazdaságossági becsléseket is kell tenni egyes jogászi hivatásokról. Tekintettel ügyvédi munkámra, a hasznosíthatóság kérdésének fókuszában is ennek a hivatásnak a sajátosságai állnak.

A dokumentum-összeállító szoftverekről

A dokumentum-összeállító szoftverek felosztását áttekinthetjük szakirodalmi vagy piaci megközelítésből is.

A dokumentum-összeállító szoftverek szakirodalmi felosztása, avagy mivel tudhat többet egy NLG-technológiát használó eszköz

A szakirodalomban a bírósági dokumentumszerkesztés kapcsán még 1993-ban Branting a dokumentum-központú megközelítéssel állította szembe a kérdésközpontú

[9] Ehud Reiter – Robert Dale: *Building Natural Language Generation Systems.* Cambridge, Cambridge University Press, 2000. 28–29.

[10] Lásd ehhez Vadász Pál – Görög György – Üveges István tanulmányának a többközpontú nyelvekre vonatkozó részét.

(issue-centric) megközelítést.[11] A dokumentum-központú megközelítés esetén Branting azt emelte ki, hogy a dokumentum valamely releváns összetevőjének (például egy bekezdésnek) a megfelelő kiválasztásán és az összetevőben elhelyezett változók felhasználói kitöltésén van a hangsúly. Sablonokból és a sablon kitöltésével kapcsolatos szabályokból áll. Branting ezzel szemben határozta meg a saját javaslatára épülő megoldást, a kérdésközpontú megközelítést: itt magának a szövegnek a megjelenési rétege mellett (fölött) kialakítanak egy olyan megjelenítési réteget, amelynek funkciója a bírói döntéshez szükséges szabályok kiértékelése, annak támogatása. Tehát a szövegezéssel együtt a bírói jogi döntés során értékelt szabályokat, a szabályokat megjelenítő jogi predikátumokat (kijelentések, állítások) is tárolják, és a „szöveg" automatizálása során mindazon szabályokat is kiértékelik, amelyek az adott döntés szempontjából relevánsak.

Ugyanezen szerző egy későbbi művében (más társszerzőkkel együtt) már az eljárási alapú (procedurális), sablonalapú és diskurzusalapú megközelítést emeli ki.[12] Az eljárási (procedurális) megközelítésben a felhasználó által a létrehozandó dokumentumra vonatkozóan kiválasztott specifikus értékek alapján egy imperatív típusú programnyelven meghatározott *utasítások választják ki* az új dokumentumhoz szükséges szövegelemeket. Idesorolták az ABF nevű rendszert, amelyhez egyébként hasonló nyelveket használnak ma is a legnépszerűbb rendszerek.

A sablonalapú megközelítés esetén a tervezők elkészítenek egy dokumentumosztályt (a „sablont"), amelyben rögzítik a sablon alá tartozó példányok közös szövegelemeit, valamint e szövegezésbe beépítik *a)* azokat a jelzőket (tokeneket), amelyek egy adott dokumentumra vonatkozó egyedi tényeket jelzik (változók, mezők), és *b)* azokat a feltételeket, amelyek meghatározzák, hogy melyik szövegrész milyen esetben illesztendő be egy-egy dokumentumba.

A diskurzusalapú dokumentum-létrehozó szoftver volt a tanulmány szerzői által követni kívánt, újnak tekintett irány, megvalósítani kívánt fő gondolat. E megközelítés célja az volt, hogy szövegfelszínen túli tartalmat is rögzítsenek, amely ez esetben a diskurzus szerkezetével kapcsolatos információk rögzítését jelentette: mi a szöveg lényege, milyen célt kívánt vele elérni a közlő (illokúciós célstruktúra, az irányadó jogi predikátumok közötti cél-alcél viszony), miként jelenik meg ez a szövegben, és mi a szöveg retorikai struktúrája (a szövegrészek között mi az, ami egymással szembe van állítva, melyik az, amelyik kifejti a másik szövegrészt, háttérül szolgál, alátámasztja a másikat stb. – ezek mindegyike egy-egy retorikai reláció, amelyekből összeadódik a retorikai struktúra).

[11] L. Karl Branting: An Issue-Oriented Approach to Judicial Document Assembly. Proceedings of the 4th International Conference on Artificial Intelligence and Law, Association for Computing Machinery. (H. n.), (k. n.), 1993. 2.

[12] L. Karl Branting et al.: Integrating Discourse and Domain Knowledge for Document Drafting. Proceedings of the 7th International Conference on Artificial Intelligence and Law, Association for Computing Machinery. Oslo, (k. n.), 1999.

Egy további, egyszerűbb szakirodalmi osztályozás a *„mail merge”* és az NLG-alapú megközelítést állítja egymással szembe.[13] A körlevél megnevezés lényege, hogy ezek a szövegszerkesztői funkciók már önmagukban is képesek összetett dokumentumautomatizálási funkciók biztosítására, így a sablonban nemcsak a változó adatok megadására (mezők), hanem feltételes szövegelemek meghatározására, hurkok definiálására is lehetőség van (például a Microsoft Word esetén a Mailings által is használt kapcsos zárójeles mezők).

A körlevél funkciót könnyű megérteni, hiszen szinte mindenki rendelkezésére áll egy ilyen eszköz, még ha nem is ismeri mindenki a felhasználásuk lehetséges komplexitását. Az NLG-megközelítést ugyanakkor nehéz jól illusztrálni, hiszen az NLG-eszközök számos, egymástól is homlokegyenest eltérő megoldásra építenek, és egyébként is szűk körben férhetők hozzá (például az egyébként sem sok helyen használt Arria vagy Yseop megoldásainak részletei a végfelhasználók számára rejtve maradnak).

Az NLG-eszközök közös pontját a legegyszerűbb talán úgy megfogalmazni, hogy van mögöttük egy közös kutatási-tudományos háttér és számítógépes nyelvészeti alap, amelyre ezek az eszközök építenek.[14] Ezzel szemben a körlevélkészítési megoldások a sablonkészítés közvetlen kihívásaira összpontosító termékek, amelyek mögött ugyanúgy lehetnek komplex és akár Turing-teljes programnyelvi megoldások, de ezek a megoldások nem fókuszálnak, reflektálnak nyelvészeti jellegű problémákra.[15] Az NLG és a pusztán sablon elhatárolásának bizonytalanságairól Reiter 2016-ban is írt egy rövid blogbejegyzést.[16]

Az NLG-eszközök a szöveggenerálási alkalmazási területeknél írtakhoz hasonlóan sokszínűek, így segíthetik chatbot működtetését vagy sablonszövegek egyedivé tételét, adatpontok szöveggé átírását vagy nagy szövegekből összefoglalók készítését is stb. Technikailag is lehet mögöttük akár egy többrétegű neurális háló architektúra,[17] de bármilyen gépi tanulási képességet nélkülöző szoftver is, amely pusztán csak egy szótárat és kézi szabályokat tartalmaz. Épülhet az NLG-eszköz olyan hagyományosan számítógépi nyelvészeti megoldásokra is, mint a szófajelemzés, hogy kiválassza a létrehozni kívánt mondatba legjobban illeszkedő szót a sok lehetséges szó közül, vagy a szóalak-generátor, hogy a megfelelő ragozási formát előállítsa.

A bemutatásukon túl a fenti felosztásokat a tanulmányban nem fogom használni.

[13] Reiter–Dale (2000): i. m. 26.
[14] Indurkhya–Damerau (2010): i. m. 121–140.
[15] Reiter–Dale (2000): i. m. 26–27.
[16] Ehud Reiter: NLG vs Templates: Levels of Sophistication in Generating Text. *Ehud Reiter's Blog*, 2016. december 18.
[17] Yang–Perlaki–Laki (2020): i. m.

A dokumentum-összeállító szoftverek piaci felosztása

Marković és Gostojić négy dokumentum-összeállító szoftvert emelt ki áttekintésében. Ezek a HotDocs, a ContractExpress, a Pathagoras és az XpressDox nevű termékek.[18] Természetesen vannak más eszközök is, sőt olyan sok, hogy gyorsan változó piacról lévén szó, szinte számba sem lehet venni mindezt. A már felsorolt négyesen túl megemlítjük pusztán példaként a Legito, a docassemble, a ClauseBase, a Woodpecker vagy a Juro termékeket is.

A termékeket áttekintve láthatunk olyanokat, ahol a dokumentum-összeállítás támogatása csak egy kis szelete a kínált funkcióinak, a fő cél inkább a szerződések kezelése (az ún. szerződés-életciklus – *contract lifecycle* – kezelése), beleértve az aláírások begyűjtését (például Juro), vagy akár a szerződés szövegének tárgyalását is, akár a szerződéskötési folyamat lezárultát követő olyan eseményekig, mint a lejárat vagy megújítás figyelése (például Coupa). Ekkor a szerkesztési, összeállítási képesség jellemzően csak egy a sok kínált funkció közül, és nem feltétlenül ez a mozzanat van a termékfejlesztés fókuszában (ilyen esetekben maga a gyártó sem feltétlenül nevezi, pozicionálja a termékét dokumentum-összeállítást támogató terméknek).

A kifejezetten dokumentum-összeállítási funkció esetén közös pontként jelölhető ki, hogy van *a)* egy megoldásuk a tervezési fázisra, a sablon létrehozására, és *b)* a létrehozott sablon használatára, kitöltésére. A tervezési fázisban is van egyrészről egy egyszerű szövegszerkesztési (vagy importálási) rész, és van az a rész, amelytől sablonná válik a szöveg: külső feltételeket, adatkapcsolatokat és kitöltendő változókat határoznak meg, valamint a szöveg megjelenésének vagy ismétlődésének szabályait rögzítik. Például egy végrendeletsablon esetében, ha lehetővé kívánjuk tenni, hogy a végrendelkező házastársak közös végrendeletet tegyenek, akkor speciális fordulatot kell a szövegben alkalmazni (például életközösség fennállása alatt tették, és egymás és a tanúk együttes jelenlétében írták alá az okiratot) – a program az előre megfogalmazott rendelkezést akkor jeleníti csak meg a szövegben, ha a sablon kitöltése során azt az instrukciót kapta, hogy erre szükség van. Ez egy kétértékű adattípus (igen/nem, azaz alkalmazandó, vagy sem), amelyet lehet, hogy a felhasználó fog megadni a sablon kitöltésekor (egy előre rögzített párbeszéd során), vagy nem párbeszédes kitöltés esetén a külső forrásból érkező adatok között szerepel a kiválasztott érték (amely lehet egy külső adatbázis, másik program vagy csak a korábban kitöltött válaszokból készített adatállomány). Ez külső feltétel, hiszen a sablontól független értékről van szó.

Általánosságban: a sablonkészítés során a szöveg elemei, változói és a külvilág közötti kapcsolatot rögzítik üzleti szabályok (üzleti „logika") meghatározásával. Lehet, hogy a sablonkészítést egy dedikált webes felületen kell megoldani (például Legito), de lehet, hogy egészében egy adott szövegszerkesztőhöz illesztett, tisztán

[18] Marko Marković – Stevan Gostojić: Knowledge-Based Legal Document Assembly. (H. n.), (k. n.), 2020. szeptember 14.

kliensoldali vagy helyiszerver-oldali megoldásban (például XpressDox, HotDocs Author, Woodpecker), vagy akár hagyományos Word-bővítményként. Persze az is gyakori, hogy a sablonszerkesztés egyes funkciói webes felületen jelennek meg (külön szerveren futva), egyes funkciói pedig a végfelhasználói gépen futva (például ActiveDocs). E funkciók megvalósításában már akár óriási komplexitással találkozhatunk, ugyanakkor az, hogy a bevezetett termék nem elég szofisztikált, sajnos jellemzően csak akkor válik nyilvánvalóvá a felhasználók számára, amikor már jó ideje használják egyszerűbb megoldásokra. Ez esetben a termék csalódást fog okozni, amely problémát tovább mélyíti, hogy az egyes termékek sablonjai között semmilyen átmenet nincsen, nincsen semmilyen szabványosítás: ha át akar térni a felhasználó az egyikről a másikra, lényegében nulláról kell kezdenie minden tervezést.

Ahhoz, hogy a rendszerek dokumentumokat tudjanak létrehozni, a sablonok mellett a dokumentumok egyedi adatait (értékeit, változóit) is be kell a rendszerbe valahogy vinni. A kitöltendő változók határozzák meg egyúttal azt is, hogy milyen, a sablonon kívüli adatforrásokból származó adatokat kell begyűjteni. Ezeket be lehet gyűjteni egy felhasználói felületre kivezetett kérdés formájában is, de ki lehet nyerni egy külső programmal létrehozott adatkapcsolatból is. A felhasználói felületre kivezetett kérdéseket hagyományosan interjúnak nevezik e körben. A terméktől függően az interjú meghatározása lehet közel automatikus (például XpressDox) vagy részletesen szabályozható (például HotDocs), beleértve például magyarázatok megadását vagy az interjúnál használt grafikai elemeket, vagy a beadott adatokra vonatkozó érvényesítési-ellenőrzési szabályokat is. Az interjú során megadott válaszokat tipikusan el lehet menteni, hogy abból újból hasonló dokumentumokat generálhassanak.

Alapvetően befolyásolja az ilyen termékek kínált funkcióit az is, hogy ki az elsődleges célközönségük. Ha a célközönség egy olyan vállalat, ahol a termék műszaki integrációja külön projekt, akkor nyilván jóval szofisztikáltabb integrációs megoldásokra lehet építeni. Ha a sablonok szerkesztése, kitöltése egy külön alkalmazott vagy konzulens feladata, akkor egészen mást értenek egyszerű sablonkészítés alatt, mint ha ugyanez egy gyakorló jogásztól vagy annak asszisztensétől elvárt művelet. Egyes termékeknél elfogadható, hogy a dokumentumautomatizálás folyamatában piaci informatikai sablonkészítő termékeket vagy programozási nyelveket használnak fel (például docassemble esetén a jinja2, Python, de ilyen a bonyolultabb interjúfunkciók JavaScript-alapú megvalósítása is). Nyilván ez utóbbi lehetőségek ugyan sokféle megoldást kínálnak, de a végfelhasználókat már csak a kész sablonokkal lehet szembesíteni. Ez esetben szinte kizárt, hogy a felhasználók maguk szerkesszék a sablont, ami egyébként a legbonyolultabb része a folyamatnak, és ami központi cél a ContractExpress, a HotDocs vagy az XpressDox esetén.

A piacon elérhető termékek többsége nincs tekintettel a felhasználás nyelvére, vagy csak néhány, nem bővíthető nyelvhez igazították. Így például olyan, egyszerűen megvalósítható, beépített nyelvspecifikus megoldásokkal találkozhatunk, mint a regionális formázási szabályok betartása, például a dátumok és a tizedesvessző/

tizedespont kérdése számítások esetén (lásd például HotDocs/LANGUAGE), de ilyen lehet a megfelelő ige- és főnévragozások nyelvi támogatása (például ClauseBase esetén az angol, francia, német és holland nyelvű igazítások, vagy csere esetén az új nőnemű alakhoz igazítja a korábban hímnemben írt főneveket is stb.).

Szintén fontos különbség, hogy egyes termékek kizárólag felhőszolgáltatásként érhetőek el (például ContractExpress, Legito), ami azt is jelenti, hogy az előfizetés megszűnése esetén az összes sablonkészítésbe fektetett energia pocsékba megy (mivel nincsen sabloncsere formátum). A felhőszolgáltatáson alapuló megoldások előnye viszont, hogy a felhasználó gyorsabban belekezdhet a használatába, és nincsen jelentős informatikai bevezetési költség. A felhős megoldások esetén az integráció biztosítása inkább az adatkapcsolatok miatt lényeges funkció, nyilván itt eleve kevesebb lehetőség van az adatkapcsolatok egyedi kialakítására, mint egy telepített *(on-premise)* megoldás esetén, ahol mélyebb tud lenni a folyamatok egymás közötti integrációja. Felhős megoldások esetén teljes egészében a gyártón múlik, hogy biztosít-e alkalmazásprogramozási felületet (*application programming interface* – API), és ha igen, milyen funkcióra (például Juro).

A piaci termékek árazását illetően a skála az ingyenes, de nehezen használható terméktől (docassemble) a nem nyilvános nagyvállalati árazásig igen széles. Vannak önálló, kis felhasználókat célzó megoldások, éves szinten ezeket már 400 $ körüli áron be lehet szerezni (WoodPecker, XpressDox), és a telepíthető (*on-premise,* nem felhős) megoldások többsége esetén nem kötelező a használathoz a továbbiakban az éves verziófrissítés, tehát elviekben csak egyszeri licenckölségeket kell érte fizetni.

A szakirodalmi felosztások helyett: milyen funkciókkal bővíthetők az összeszerkesztési megoldások?

Nézzük meg, hogy milyen területen lehet bővíteni az összeszerkesztési megoldások funkcióit! Ehhez először áttekintjük a szakirodalmi megkülönböztetés okát és indokait, majd azt, hogy milyen szövegen túli funkciókban lehet egyáltalán gondolkozni.

Az idézett művek a fenti felosztásokkal igazából csak arra hívták fel a figyelmet, hogy vannak olyan speciális, nyelvészeti alapú szoftverfunkciók, amelyek az egyes szöveggenerálási felhasználások szempontjából előnyösek (mintha az idézett művek saját maguk létjogosultságát kívánnák alátámasztani a felosztás megfogalmazásával.) Az eljárási és a sablonalapú megközelítés lényegében egyidejű a számítógépes szövegfeldolgozás képességével (számítógépes makrónyelvek, sablonkészítő motorok, adatbázisokhoz való hozzáférésre vagy változókra épülő szövegbeillesztési képességek stb.), de a nyers szövegben már itt is megjelennek olyan szimbólumok, amelyek a feldolgozás vagy felhasználás módjára vonatkozóan adnak utasításokat.

Vegyük sorra, hogy az idézett szakirodalmi szerzők mit emelnek ki az NLG-megoldások előnyeként! Branting a saját megközelítés előnyeként kiemelte a módszer rugalmasságát: azt, hogy a bírói döntések támogatására is lehet hasznosítani,

a döntés megmagyarázásában is segít, és könnyebb a karbantartása.[19] Későbbi tanulmányában azt emelte ki, hogy bizonyos felhasználások esetén szükség van arra is, hogy a felhasználó bármikor megnézhesse, miért is került bele az adott szöveg a tervezetbe, mi okozta a változást, és módosíthasson is ezen, ha szükséges *(„queriable liveliness").*[20] Ebben a tanulmányában a kereskedelmi forgalomban kapható eszközökkel szemben azt a kritikát fogalmazta meg, hogy azok a nyelvtől és jogtól (azaz tudásterülettől) független megoldások, és ez a felhasználásukat rugalmatlanná teszi. Már ekkor hangsúlyozottan felhívta a szerző a figyelmet arra, hogy az NLG-kutatások milyen sokat tudnának segíteni ezen a területen, idézve Reiter egy korábbi művéből:[21] könnyebb a szöveg karbantarthatósága, egyszerűbbek az idővel szükségessé váló módosítások, frissítések, mindez jobb minőségű, érthetőbb szövegeket eredményezhet, amelyek a mondatokon átnyúló viszonyokat is tükrözni és követni tudják (például az említett retorikai szerkezetet), ezáltal könnyebb például a több nyelvre történő adaptációjuk vagy a megfelelőségi (compliance) szempontokból történő ellenőrzésük is.

Egy évvel később Reiter és Dale e tárgyban általában a gépi és az emberi szöveggenerálás összehasonlítása során (tehát nem a sablonalapú és NLG-alapú dokumentum-összeállító megoldások összevetése során) lényegében ugyanezeket az előnyöket emelte ki:[22] a szöveggeneráló megoldások az emberiekhez képest nagyobb konzisztenciát tudnak biztosítani, szabványosabb végeredményt hoznak létre, gyorsabban és több nyelven. Ugyanez a szöveg azonban azt is hangsúlyozza, hogy az NLG-megoldások fejlesztései akkor térülnek meg, ha kellően nagy mennyiségben állítanak elő dokumentumokat[23] – nyilván azóta megváltozott az egyes megoldások fejlesztésének költsége, de ezt a szempontot nem szabad figyelmen kívül hagyni.

A felsorolt előnyök forrását nézve alapvető, hogy a diskurzusalapú és a kérdésközpontú megoldások esetén a „nyers szöveg" mellett (azon túl) olyan pluszinformációkat is rögzíteni szeretnének a szerzők, amelyek a felhasználási folyamatban az adott felhasználási területtől függően hasznosak lehetnek.

Így például a kérdés- vagy diskurzusközpontú megközelítés használata kifejezetten bírócentrikus felhasználásról szól, egy másik tanulmányban hasonló célból egy informatikailag leírt (ábrázolt) jogi szabályrendszer határozza meg, hogy a felhasználónak az interjú során milyen kérdésekre kell választ adnia.[24] Nyilván egy ilyen rendszer vagy annak egyes részei (például a gépileg leírt szabályrendszer tudásbázisa) nem célszerű, ha mondjuk ügyvédek vagy más felhasználók esetén is

[19] Branting (1993): i. m. 3.
[20] Branting et al. (1999): i. m. 2–3.
[21] Ehud Reiter: NLG vs. Templates. *Proceedings of the Fifth European Workshop on Natural Language Generation,* Leiden, (k. n.), 1995.
[22] Reiter–Dale (2000): i. m. 29–30.
[23] Reiter–Dale (2000): i. m. 28–29.
[24] Marković–Gostojić (2000): i. m.

változatlanul hozzáférhetők, ahogy a gépi következtetések levonásának és az attól való eltérésnek is egész más a feltételrendszere egy automatikus döntéshozatalhoz csatlakozó hatósági felhasználás során, mint egy büntetőjogi ítélet esetén. Mindegyik hivatkozott megoldás épít tehát valami szövegen túli adatra: lehet az egy gépileg feldolgozható formában rögzített szabályrendszer, egy retorikai struktúra (DocuPlanner)[25] valamilyen, a szöveg tárgyával kapcsolatos szemantikai leíró tartalom vagy metaadat, vagy akár egy általánosabb jogi tudásleírás.[26]

Lauritsen a dokumentumkészítést támogató rendszerekkel kapcsolatos általános elméleti vázlatában számos szövegen túli tényező megjeleníthetőségét, rögzíthetőségét is említi,[27] így a fenti szemantikai elemeken túl idesorolja például a magyarázatokat, a szöveg forrását, a feltételes vagy ismétlődő szövegeket, a kötelező és lehetséges sémaelemeket, valamint ilyen leírandó elemként emeli ki a teljes jogi praxison belüli szövegek közötti egyértelmű hivatkozhatóságot vagy a külső hivatkozásokat is.

A dokumentum-összeállító eszközök képességeinek leírása szempontjából tehát leírhatunk nyelv- és jogi tudástól független technikai funkciókat, de ez a technika már a gépi szövegfeldolgozás kezdete óta rendelkezésre áll, azaz a számítógéppel lényegében egyidős, a jogászok számára pedig legalább 1979 óta (ABF) dedikált piaci termékként is elérhető, de a legtöbb szövegszerkesztő is készen biztosít ilyen funkciókat. Ilyen, a használt nyelvtől független funkciók a már ismertetett szövegváltozatok használata (felsorolásban vagy egész mondatokban) vagy a felhasználó által megadott egyedi értékek (például számok) beágyazása a sablon véglegesítése során. Ilyen az adatbázis-kapcsolatok használata is, azaz amikor egy szövegrész helyettesítő értékét mondjuk egy külső munkavállaló- vagy partnernyilvántartásból veszik (mint egy klasszikus körlevélkészítési funkció esetén, ahol a partneradatbázisból kiválasztják a címzett partnereket és képviselőjüket, azok címét stb.). Természetesen vannak helyzetek, amikor az ilyen nyelvfüggetlen helyettesítés önmagában nem elegendő, hiszen nem teljes mondatok közül kell választani, hanem szavak közül, ahol a megfelelő forma már mindig nyelvtani illesztést igényel.

A szöveggenerálás adott (ügyviteli) folyamatát tehát a hagyományos, nyelvfüggetlen technikai megoldásokon túlmutatóan is számos területen lehet támogatni. Kérdés, hogy ilyen esetben az eltérő felhasználási esetek eltérő terméket indokolnak-e, mert ha igen, ez jelentősen csökkenti a valószínűségét annak, hogy egyes összeszerkesztési termékek széles körben használt megoldásokká válnak.

Minél sokoldalúbb felhasználásra terveznek egy ilyen megoldást, annál bonyolultabb lesz magát az automatizálást megvalósítani és felhasználni. Ez nem egyszerűen azon múlik, hogy egy felhasználói felület kortárs grafikai tervezői hagyományokra

[25] Branting et al. (1999): i. m.
[26] Kilián Imre: Szemantikai alapú jogi tudás-menedzsment technológiák. PhD-értekezés. Pécs, Pécsi Tudományegyetem Állam- és Jogtudományi Kar, Doktori Iskola, 2013. 150–156.
[27] Lauritsen (2007): i. m. 10.

és felhasználói felületi standardokra épít-e, vagy sem. A probléma az, hogy egy dokumentum összeszerkesztése mögött igen komplex döntések vannak, amit egyszerűen nem lehet áttekinthető, de egyúttal flexibilis módon rögzíteni (lásd például a Legito sablonszerkesztési felületét – modern, de semmivel nem átláthatóbb, mint a harmincéves HotDocs megoldása, viszont jóval kevésbé tud rugalmas lenni).

Néhányan próbálkoznak ún. *„no code"* megoldásokkal is, hogy egy ábrán lehessen kattintgatni és válogatni a szükséges megoldásokat (lásd például a Google Blocklyhoz hasonló megoldásokat). Ez nem oldja meg a dokumentumok összeállításának komplexitását: a feladat nem lesz sokkal könnyebb attól, hogy nem egyszavas parancsokat kell megtalálnunk és beírnunk, hanem az ilyen tartalmú parancsokat vizuálisan megjelenítő blokkokat húzogatjuk, ágyazzuk egymásba.

A dokumentum részeinél megjelenő összetettség nem abból eredő probléma, hogy egyes információ vagy parancs miként jelenik meg a felhasználó számára, tehát nem elsősorban a megfelelő felhasználói felület hiánya miatt lesz nehéz a kódolás, programozás. A probléma sokkal inkább a megfelelő absztrakció és modularitás hiányából fakad, hiszen túl sok, a jogászi gondolkodáson túlmutató, a jogászi gondolkodás szempontjából irreleváns részletre kell odafigyelni, a figyelem aránytalanul nagy részét ilyen, jogi szempontból mellékes kérdések rendezésére kell fordítani.

Például egy szerződést szerkesztő jogász számára mindegy, hogy a fizetési kötelezettséget meghatározó rendelkezést vállalkozási vagy adásvételi keretszerződésnél kívánja-e alkalmazni, nem arra kellene fókuszálnia, hogy éppen vállalkozónak, eladónak vagy megbízottnak hívják-e a feleket, de arra üzleti szempontból nagyon oda kell figyelni, hogy előre vagy utólag történik-e a fizetés, illetve a fizetési határidő 15 vagy 120 nap, mert más-más üzleti kockázatokat kell kezelni.

A kis ügyvédi irodák számára jelenleg forgalmazott piaci termékek többsége nem rendelkezik olyan funkciókkal, amelyek ezeket a szempontokat támogatják, hiszen azok a hangsúlyt az angol nyelvű szöveg beillesztésére vagy ismétlésére helyezik. Emiatt pedig eltérő változatokat kellene megírnunk minden szövegből megbízási, adásvételi, forgalmazási stb. szerződések esetén. Még ha az ügyvédi végtermék leggyakrabban szöveg is, a szöveg csak a jogi tudáson és nyelvi jellegű (mentális) eszközökön alapuló, magasabb szintű feldolgozás végső kimenete, és a kettő közötti feszültség okozza azt a komplexitást, költséget, amely nehezíti az ilyen eszközök elterjedését.

Ha a sablonbeli szöveg nyelvi rétege nem kellően absztrakt, akkor a nyelvi illesztéseket üzleti logikai eszközökkel kell megpróbálni pótolni: külön változókat kell rögzíteni a szövegben, hogy ha mondjuk megbízási szerződésről van szó, akkor ne eladót, hanem megbízottat írjon. Az egyes szavakat és toldalékokat mindig megfelelően illeszteni kell a szöveghez, azaz ha eladóról van szó, akkor minden olyan szót, ahol eladó van, „az" névelőnek kell megelőznie, míg vállalkozó esetén „a" névelőnek stb. Mindez fölöslegesen lelassítja a szövegsablonok létrehozását is, és ezáltal megemeli az automatizálás költségét is.

A jogászi logika szerint nem elegendő az egyes önálló szerződések szintjéhez illeszkedő sablonokban gondolkozni, hanem minél általánosabb szintű rendelkezésekre, klauzulákra van szükség. Csak így lehet biztosítani, hogy a szerződés típusától függetlenül ahol indokolt, ugyanazokat a rendelkezéseket használják, mivel a hasonló jogi célokat indokolt hasonló szövegezéssel megoldani. Fontos lenne, hogy a legegyszerűbben, de a legtöbb helyzetben alkalmazható módon lehessen az összeszerkesztési eszközben rögzíteni minden szövegváltozatot. Az automatizálás költségei azonban tovább fokozódnak, ha a nyelvi rétegen túl olyan tudással kapcsolatos információkat is szeretnénk rögzíteni a dokumentum-összeszerkesztési megoldásban, mint a szöveg mögötti jogi (vagy más szemantikai) tartalom. Például ha egy évekig elnyúló teljesítési idejű ingatlan-adásvételi szerződést kötünk, akkor – amíg az 1997. évi törvény hatályos – nem elég a tulajdonjogot fenntartani (és az elidegenítési és terhelési tilalmat biztosítani), ne felejtsük az első vételárrészt zálogjoggal biztosítani, sőt, a programnak elállási jogot is kellene ilyen esetben rögzítenie, ha egy hatóság időközben végrehajtási jogot jegyezne be az ingatlanra.

Ez a réteg az üzleti vagy jogi logika szöveg mögötti rétege. Ezeket az extra rétegeket is lehet valahogyan rögzíteni, gyűjteni, csak elég bonyolult, mondatokon és bekezdéseken túlnyúló összefüggéseket kell lejegyezni, amelyek egymásnak is ellentmondhatnak. Ezeket az összefüggéseket, kapcsolatokat valahogyan le kell írni, akár a sablon létrehozásakor egy szöveges útmutatóval magyarázva, akár az egymásnak ellentmondó opciókat kizáró menük létrehozásával, akár egy gépileg is közvetlenül feldolgozható leíró nyelvvel.[28] Az összeszerkesztő így segíthet abban, hogy a megadott „jogi és üzleti" szempontrendszerből megfelelően válogasson, priorizáljon, rögzítsen előre kódolt mérlegelési helyzeteket. (Ez volt a korábban diskurzusalapúként leírt rendszerekben is az újdonság.)

Kevesen használtak ez idáig jogi tudásleírási eszközt. Aki rendelkezik is ilyen tapasztalattal, az is csak egy-egy konkrét megoldást ismer, és jó eséllyel sok mindent újra kell tanulnia új felhasználás esetén. Tehát minél sokoldalúbb a felhasználás, annál nagyobb lesz sajnos az automatizálás költsége is. Az automatizálás költségeit tovább növeli a szűk felhasználói bázis, a termékek közötti standardizálás és adatcsere hiánya, a szerteágazó funkciók és a nyelvenként eltérő igények piaca. Sajnos amíg ez a standardizálás nem valósul meg, és a kisebb felhasználók által is elérhető piac nem konszolidálódik, addig az automatizálás jogi iparági költségei nem fognak csökkenni, így előfordulhat, hogy ilyen támogatás hiányában nem éri meg egy adott felhasználási területre tartozó dokumentum létrehozásának, összeállításának támogatása, automatizálása.

[28] Lásd a Marković–Gostojić (2000): i. m. 4–6. oldalon olvasható szabályleírásokat LegalRuleML formátumban leírva, amelyekből az interjú során adott válaszok alapján a rendszernek kell eldöntenie, hogy mi legyen a releváns következtetés a tényállásról és a jogcímről, és annak megfelelően szövegezze meg a vádirat vázlatát.

A szövegkivonatoló eszközök és általában a többrétegű neurális modellek jogi felhasználhatóságáról

A szöveggenerálásra használt eszközök köre nem merül ki a dokumentumösszeállítási termékekben, ezért érdemes röviden kitérni arra is, hogy a jogi felhasználások számára milyen lehetőséget kínálnak a szövegkivonatoló eszközök. A vizsgált konkrét tanulmány egy magyar nyelvű felhasználásra vonatkozott, de ez jól példázza azt is, hogy más, hasonló méretű nyelveknél hasonló eredményekre lehet számítani.[29] Ez esetben a generált szöveg ún. végponttól végpontig neurális modellen alapult (azaz egy klasszikus, természetes nyelvi feldolgozási feladatsor [pipeline] valamennyi összetevőjét helyettesítette).

A hivatkozott tanulmány bemutatta, hogy milyen eredménnyel tudták a BERT többnyelvű változatának az adott kis nyelvre vonatkozó változatát felhasználni arra a célra, hogy egyes magyar nyelvű cikkekből egy pár mondatos összefoglalót készítsen. Végtelenül leegyszerűsítve a folyamatot: a BERT egy olyan, a Google által kifejlesztett és közzétett, többrétegű neurális hálóra épülő nyelvi modellarchitektúra, amelyet igen sokféle módon használnak nyelvi gépi tanulási feladatokra. Az egyik több nyelven tanított, a Google által már előre betanított modellt hasznosították újra a kutatók (BERT-Base, Multilingual Cased, máshol multi-BERT néven hivatkozzák). A kutatók egy összegzési feladatra igazították, „finomhangolták" ezt, és kísérletezgettek a legjobb eredményt biztosító architektúrával (például az ún. extraktív modell esetén bemeneti oldalra elküldték az összefoglalandó cikk egyes mondatait [azok numerikus megjelenítését], és a modell feladata volt helyesen megtippelni, hogy mi a tartalmat legjobban leíró három mondat, majd a tanítás részeként összevetették, hogy a modellből kijövő végeredmény mennyiben hasonlít a kézzel írt összefoglalókhoz, és így visszacsatolva javították a modellezést). A végeredmény a legjobb architektúra esetén 55,46%-os „találat" lett (szavak azonosságának mértéke, ún. unigram fedés). Hogy ez jogi célra mennyiben használható, erre később még visszatérek.

Más kutatók bemutatták, hogy jobb eredményt is el lehet érni bizonyos területeken, ha a több nyelven tanított és közzétett Google-modell helyett ugyanazon módszer szerint elvégeznek egy önálló tanítást kizárólag az adott (jelen esetben szintén magyar) nyelvi korpuszra, szöveganyagra alapozva (új előtanítással). Az egyik kutatás egy kisebb modell tanítását mutatta be (huBERT),[30] egy másik kutatás pedig

[29] Yang–Perlaki–Laki (2000): i. m.

[30] Dávid Márk Nemeskey: *Introducing HuBERT*. XVII. Magyar Számítógépes Nyelvészeti Konferencia. Szeged, Szegedi Tudományegyetem Természettudományi és Informatikai Kar, Informatikai Intézet, 2021.

a nagyobb méretű szövegen elvégzett tanítást részletezte (HILBERT),[31] ez utóbbi előtanítás a jelenlegi árakon kb. 6500 $ költséggel járt, és 24 GB bemeneti adatot igényelt. Kizárólag a magyar nyelvű korpuszon tanított modell egyértelműen jobb eredményt adott a többnyelvű változathoz képest.[32] Hozzá kell tenni, hogy laikus (ügyvédi) szemmel nézve meglepő, hogy egy többnyelvű modellel is milyen jól használható szövegeket, eredményeket lehet létrehozni.

Míg a magyar nyelvű (általános) korpuszon tanított modell használatával jelentős javulást lehet elérni a többnyelvű (általános) modellhez képest, addig a kifejezetten jogi korpuszon való tanítással elérhető javulás jóval mérsékeltebb az általános magyar nyelvi modellel elérhető eredményekhez képest. Angol nyelven közzétettek egy olyan tanulmányt, amely azt mutatja meg, hogy el lehet-e érni bizonyos feladatok terén nagyobb pontosságot (jobb eredményt), ha már magát az előtanítást is jogi szövegeken végzik, összehasonlítva azzal, ha csak a finomhangolást végzik valamilyen jogi szövegen (LegalBERT).[33] Az előtanítások vagy finomhangolások alapjául 1,9 GB-nyi EU-s jogi aktusok (EURLEX), 0,6 GB bírósági döntések, illetve más hasonló adatok szolgáltak, például az USA Securities Exchange Commission EDGAR-féle adatbázisa (3,9 GB). A felmérést olyan jogi természetű szövegmegértési feladatokon hajtották végre, mint például ki a szerződő fél, mi a szerződés címe, mi az irányadó jog vagy joghatóság, vagy melyek a bérleti szerződés kulcselemei a szövegben (például ki a bérbeadó, a megszűnés napja, határozott időtartam, a bérleti díj mértéke).[34] Az eredmények kismértékben javultak úgy, hogy minél nehezebb volt a feladat, annál nagyobb mértékű volt ez a javulás. Azonban összességében a kifejezetten jogi szövegre épített tanítás nem eredményezett a jogi munka szempontjából jelentős javulást egyik feladatnál sem. A legnagyobb mértékű javulást ez a fajta tanítás a bírósági ítéletek címkézése, kategorizálása terén érte el, és a javulás mértéke ott is csak 2,5 százalék volt. Hozzá kell tenni, hogy ezeknél a jogi feladatoknál gyakran nem a BERT-modell adta a legjobb eredményt, hanem más többrétegű neurális háló architektúrák (például a bérleti szerződéseknél az esetek 87 százalékában megtalálta a helyes rendelkezést).

Ez számunkra mindössze illusztrációként fontos: a többrétegű neurális háló modellekre épülő megoldások a jogi szövegösszegzés terén is ígéretesek lehetnek, azonban ezek használata egy adott tagállami kis nyelven is további befektetést és kutatást, fejlesztést igényel. A befektetés jelentheti azt is, hogy egy adott nyelven

[31] Feldmann Ádám et al.: *HILBERT, magyar nyelvű BERT-large modell tanítása felhő környezetben*. XVII. Magyar Számítógépes Nyelvészeti Konferencia. Szeged, Szegedi Tudományegyetem Természettudományi és Informatikai Kar, Informatikai Intézet, 2021. 29–36.

[32] Nemeskey (2021): i. m.

[33] Ilias Chalkidis et al.: *LEGAL-BERT: The Muppets Straight out of Law School*. Conference: Findings of the Association for Computational Linguistics: EMNLP 2020. (H. n.), (k. n.), 2020. október 6. 2898–2904.; Ilias Chalkidis et al.: Neural Contract Element Extraction Revisited: Letters from Sesame Street. (H. n.), (k. n.), 2021. február 22.

[34] Chalkidis et al. (2020): i. m. (32. lj.)

jogi szövegeket kell a tanításra összegyűjteni és előkészíteni. De maga a tanítás is költséget jelenthet, amellyel számolni kell. Azonban ennek is vannak olyan egyéb előfeltételei, mint a tanításhoz szükséges, emberi ráfordítást is igénylő kidolgozott tanítókészlet (például 2080 + 289 példa az irányadó jogra, 2552 példa a bérleményre).[35] A továbblépésben nem okozhat általában gondot a rendelkezésre álló joganyag mérete sem (összehasonlításként: a körülbelül 170 ezer anonimizált magyar bírósági ítélet átlagos hosszával számolva a magyar bírói joganyag mérete meg is haladja a rendelkezésre álló uniós bírói ítéletek méretét, tehát ez hasonló méretű joganyaggal rendelkező más országokban sem lehet nehézség).

Ugyanakkor a rendelkezésre álló adatméret bizonyos felhasználásoknál még komoly gyakorlati problémát okozhat. Nyilván a SEC-féle EDGAR szerződéses adatállomány egyszerűen nem áll rendelkezésre a kis nyelveken akkor sem, ha mindenki bedobná az összes saját szerződését egy közös korpuszba (ami persze nem is lenne jó ötlet, hiszen a nyelvi modellekből igen érzékeny információkat is vissza lehet fejteni).[36] Érdemes odafigyelni arra a hasonló figyelmeztetésre is, hogy néhány fontos területen a gépi tanulással kapcsolatos kutatások a jobb eredményeket egyre nagyobb modellekre építve érik el, ami nemcsak a paraméterek számának növekedésével jár, hanem a tanításhoz szükséges tanító adat mennyiségének növekedésével is. Ilyen például a GPT-3 (Generative Pre-trained Transformer) fantázianevű nyelvi modell, amelyet elsősorban az ember által írtakhoz hasonló szöveg generálására használnak.[37] A GPT-3 nyelvi modell tanításához 45 TB-nyi adatot használtak – nyilván ilyen méretben magyar vagy más, nem világnyelvű, ember által létrehozott szöveganyag nemhogy nincs, de várhatóan soha nem is lesz. (Csak a nagyságrendek érzékeltetéséhez: a teljes angol nyelvű Wikipédia kitömörítve és képek nélkül jelenleg kb. 70 GB körüli méretű, és 6 362 952 szócikket tartalmaz, a magyar Wikipédia pedig 491 227 szócikket közöl.) A tendenciát aztán más modellek is követték, bár a tanító anyag méretéről nem mindig közölnek információt.[38]

Egy tanulmány szerint kb. 10-100 millió szavas tanító anyagra van szükség ahhoz, hogy valamely modell megtanulja a nyelvi szintaktikai és szemantikai jellemzők többségét, de ennél jóval nagyobb méretű anyagra van szükség ahhoz, hogy az olyan egyéb, a nyelvben is megjelenő háttérismereteket felismerhessék, mint amilyeneket például „józan ész" alá szoktak sorolni.[39] Szintaktikai típusú feladat

[35] Uo.

[36] Nicholas Carlini et al.: Extracting Training Data from Large Language Models. (H. n.), (k. n.), 2021. június 15.

[37] Lásd https://github.com/openai/gpt-3.

[38] A paraméter modellszámból nem vonnék le önmagában ilyen következtetést, de a GPT-3 nyelvi modell 175 milliárd [10^9] paraméterszámát legutóbb a Google Switch haladta meg 1,6 billió [10^{12}] paraméterrel. William Fedus – Barret Zoph – Noam Shazeer: Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. (H. n.), (k. n.), 2022. június 16.

[39] Yian Zhang et al.: When Do You Need Billions of Words of Pretraining Data? (H. n.), (k. n.), 2020.

például az, hogy megtalálja egy szó vagy mondatrész nyelvtani szerepét (például szófaját vagy a főnév a bővítményét stb.), az ilyen feladatnál a hivatkozott tanulmány már 20 millió szóval való tanítás után sem javult érdemben. Ezzel szemben az olyan feladatoknál, ahol a helyes megoldáshoz két, nyelvtanilag ugyanúgy helyes szó közül azt kellett kiválasztani, amelyikkel értelmessé válik a mondat, még a 30 milliárd szavas tanítási anyaggal is további jelentős javulást lehetett elérni.

Ez alátámasztja, hogy a nyelvi modelleknek milyen meghatározó szerepük van a jogi jellegű nyelvi feladatok megoldásában is, és azt is, hogy számos, főleg szöveggenerálási feladatnál miért nem lehet pusztán a GPT-3 jellegű modellekre bízni a megfelelő kialakítást.

A jogi nyelvi felhasználás még azonos nyelveken belül is eltérő lehet (például az osztrák és a német jogi nyelv eltérése), nem reális, hogy ilyen GPT-3 méretű adatkészleten tanított modellekkel akár közepes méretű nyelveken is rendelkezzünk, főleg nem kifejezetten a jogi területen.

A jogi területen belül kiemelt fontosságú szempont a nyelvi pontosság, ami nem minden felhasználási területen jellemző. Ha nagyszámú szerződés tartalmának összefoglalásáról vagy elemei kivonatolásáról van szó, és a kockázatot a felmérést kérő ügyfél elfogadja a gyors vagy olcsóbb felmérés fejében, akkor itt valóban számos célra elegendő lehet egy például 70 százalékos pontosságú felismerés (de egy 60 százalékos már nem biztos). Ha egy szövegkinyerésre használt megoldás csak támogatja a jogász munkáját, és a vélt találatokat meg tudja mutatni a szövegkörnyezetében a felhasználónak, mielőtt azt elfogadják eredményként (például Kira Systems), szintén nagy segítség lehet egy 80 százalékos pontosságú keresés is.

A dokumentum-összeállítási felhasználás szempontjából azonban ez a fajta nagyvonalú megközelítés problematikusabb vagy akár elfogadhatatlan – ahogyan egy orvosi adatból szöveget alkotó rendszer esetén sem szerencsés, ha az egyik betegség helyett egy hangzásában nagyon hasonló betegséget jelenít meg a végponttól végpontig neurális hálós modellre épülő eszköz.

Ha rövid a létrehozott szöveg, és azt a munkafolyamatban megnézi egy, a hiba észlelésére kellően felkészült ember, a kisebb hibák nem okoznak gondot. De ha a generált szöveg hosszú, a potenciális hibaforrás pedig számtalan szöveghelyen megjelenhet, akkor az egész szövegautomatizálási megoldás értelmét veszítheti, hiszen a javítással több időt kell eltölteni, mint ha teljesen manuálisan hoznánk létre a szöveget. Ugyanígy veszélyes, ha a hibát csak a jogi-gazdasági kontextust értő személy tudja észlelni, mondjuk egy megbízó-megbízott szócsere esetén.

Ehud Reiter tárgybeli blogbejegyzései[40] szerint a szöveggenerálási feladatok esetén a szöveg pontossága szerint hátrányban vannak az olyan technikára épülő modellek, amelyek a nem nyelvi adatból közvetlenül egy neurális háló segítségével

[40] Ehud Reiter: Generated Texts Must Be Accurate! *Ehud Reiter's Blog,* 2019. szeptember 26.; Ehud Reiter: Academic NLG Should Not Fixate on End-to-End Neural. *Ehud Reiter's Blog,* 2020. december 1.

építenek természetes nyelvi szöveget (mint a jelen pontban írt példák) ahhoz képest, amikor a neurális modellek a természetes nyelvi szövegalkotási feladatsor *(pipeline)* egyes funkcióit látják el. A szövegalkotási feladatsor például a kívánt szöveget úgy építi fel, hogy az elvárt tartalmat leíró paraméterek alapján kiválasztja a szövegkörnyezetben megfelelő mondatszerkezeteket, kifejezéseket és szavakat, majd azokat nyelvtanilag megfelelően egymáshoz igazítja, így a neurális hálón alapuló modellek segítségével helyes morfémákat hoz létre.[41]

Milyen eszközök elvi felhasználását érdemes megvizsgálni vagy fejleszteni a kisebb nyelvet használó országokban?

A fentiekből láthatjuk, hogy egyes kutatási és fejlesztési feladatok számos terület jogi felhasználójának is segíthetnek, mondhatni összjogászi érdekeket képviselnek, míg egyes feladatok konkrétan egy-egy szűkebb felhasználói csoport számára kínálnak csak megoldást.

Akármelyik hivatásról is legyen szó, minden felhasználást erőteljesen segít, ha minél több, megfelelően anonimizált jogi természetű adat áll rendelkezésre. Legyen az bírósági vagy valamilyen hatósági döntés, elvi jellegű vagy egyedi jellegű, elektronikusan elérhető aktuális vagy már archív jogirodalom, hatályos vagy már hatálytalan normaszöveg – a rendelkezésre álló és előállítható adatok körének felmérése az első lépés ahhoz, hogy megbecsüljük, milyen AI/NLP-felhasználások lehetnek reálisak és milyen költségen. Ez minden további munka alapfeltétele, legyen szó nyers tanítási anyagról (például egy jogi nyelvi modellhez szükséges adatkészletről) vagy kézi tanítási anyag előállításához használt alapanyagról. Az újraazonosítás kockázata ellenére is hasznos lenne egy nagy szerződéses szövegadatbázis is, akár állami és önkormányzati szerződésekből.

Kétségtelenül a jogi felhasználás is sokat profitálna abból, ha az adott ország hivatalos nyelvének számítógépes nyelvészeti kutatásait támogatják, és ez kiemelten sokat segítene a szöveggenerálás terén, hogy mindenki számára elérhető szabványos megoldások álljanak rendelkezésre. Ilyen például a SimpleNLG,[42] amely egy Java nyelven írt alkalmazásprogramozási interfész. Képes arra, hogy a külön megadott alany, állítmány és tárgy szavaiból megfelelő mondatot hozzon létre. Például a „NPPhraseSpec subject = nlgFactory.createNounPhrase(„Mary”); NPPhraseSpec object = nlgFactory.createNounPhrase(„the monkey”); VPPhraseSpec verb = nlgFactory.createVerbPhrase(„chase”);” verb.addModifier(„quickly”); parancsokból kihozza a nyelvtanilag helyes „Mary gyorsan üldözi a majmot” mondatot az adott nyelven,

[41] Thiago Castro Ferreira et al.: Neural Data-to-Text Generation: A Comparison between Pipeline and End-to-End Architectures. (H. n.), (k. n.), 2019. november 27.
[42] Lásd: https://github.com/simplenlg/simplenlg.

vagy egy megadott mondatot nyelvtanilag helyesen át tud alakítani grammatikai tagadó alakba.

Nem az a cél, hogy a jogászok maguk közvetlenül használják az olyan eszközöket, mint a SimpleNLG, de ha ilyen eszközök elérhetővé válnak a jogi hivatásokat kiszolgáló informatikai vállalkozások részére, akkor ez csökkenti e vállalkozások költségeit, és ezáltal növeli annak a valószínűségét, hogy a jogi hivatásbeliek igényeit megpróbálják kiszolgálni. A szabványosított megoldások támogatása jelentheti a rendelkezésre álló nyílt forráskódú eszközök fejlesztését vagy bővítését, az ilyen eszközök egyes funkcióinak bővítését, a meglévő eszközök új nyelvekhez való hozzáigazítását, vagy akár csak az ilyen eszközök adaptációs költségeinek csökkentését új integrációs módozatok, másik nyelvhez vagy programhoz való illesztések biztosításával. A jogi hivatás gyakorlóira specializálódott informatikai vállalkozások (az ügyfélkezelési vagy praxiskezelési szoftverek kiadói) nem feltétlenül NLP-specialisták is egyben, így néha akár az is segíthet, ha a felhasználók képviselői felhívják a figyelmüket ezekre a lehetőségekre.

Bár nem a nyelvgenerálással kapcsolatos hiány, de a kisebb nyelveken elérhető, jogi tárgyú NLP-eszközök képességein nagyot tudna lendíteni néhány nyilvánosan hozzáférhető, adott nyelvű tanító (értékelő) adatkészlet, amely jogi tárgyú, és egyébként kézi feldolgozást igényelne. Ilyen lehetne például a bérleti vagy adásvételi szerződéses elemeket beazonosító tanítókészlet.[43]

Mind a szöveggenerálás, mind a szövegelemzés szempontjából jelentős fejlődést lehetne elérni számos területen további sablonszövegek részletesebb kidolgozásával, a jelenlegi sablonok elmélyítésével is. Ilyen lehetséges terület lehetne például a perjogi beadványoknál tipikus modulok, szövegrészek kidolgozása, hasonlóan a Casetext Compose termékhez, ahol joghatóságonként és jogi területenként aprólékos szerkesztői munkával kidolgozták a tipikus indítványok szövegezését (például szakértő kizárásának szövegezése). Ez a jelentős erőfeszítést igénylő munka egyrészről a szöveggenerálási feladatokat is segíti, mert van standardizált szöveg, amelynek megjelenését hozzá kell igazítani az adott szövegkörnyezethez, ugyanakkor egy-egy ilyen szövegrész elhelyezése a beadvány vagy határozat magasabb szintű struktúrájáról is lényeges információt rögzít, és a hasonló tárgyú indítványokkal kapcsolatos döntések keresését is elősegíti. Az ilyen jellegű felhasználásnak nemcsak ügyvédi, hanem bírói, közjegyzői oldalon is van haszna.

Nehéz elképzelni, hogy a szövegszerkesztőket ki lehet robbantani a felhasználás középpontjából, ennek ellenére hasznos lenne azt is támogatni, hogy a hétköznapi felhasználók a szöveggel együtt képesek legyenek olyan szövegen túli részletek rögzítésére is, mint a szöveghez kapcsolt szemantikai tartalom. Ez jól hasznosítható lenne szerződéseknél, beadványoknál, határozatoknál és szövegtáraknál (rendelkezéstáraknál) is. Az így rögzített adatokból kellő általánossággal lehetne

[43] Chalkidis et al. (2020): i. m. (33. lj.)

szintaktikailag is helyes szöveget generálni (például tudásreprezentáció RDF-ben és szöveggenerálás RDF-to-text technikákra építve).

A szöveggenerálás néhány eszköze kétségtelenül a jogi felhasználás szempontjából marginális jelentőségű, például a kérdés-felelet típusú rendszerek és a chatbotok. A szövegkivonatolás azonban bizonyos területeken hasznos nemcsak az ügyvédi szakma képviselőinek, de a nagy adatmennyiséget rövid időn belül feldolgozni kívánó ügyészi, rendőrségi (de akár bírósági) dolgozóknak, és idesorolhatjuk például a felszámolókat is. A szöveggenerálási eszközök közül pedig a dokumentumösszeállítási megoldások azok, amelyek szinte minden szakma számára segítséget tudnának nyújtani, csak éppen kissé eltérő felhasználói felületen, más szövegek és rendelkezések automatizálása terén.

Sajnos máig ugyanúgy időszerű maradt az a kérdés, amelyet még 1979-ben tettek fel az American Bar Foundationnél: „Egy ügyvéd, programozó segítsége nélkül, létre tud-e hozni és fenn tud-e tartani olyan számítógépes rendszert, amely segít lefolytatni az interjút, jogi következtetéseket von le, és összerakja a megfelelő jogi űrlapot? Több ügyvéd használhat együtt egy ilyen rendszert?"[44]

A nehézségek ellenére látni kell, hogy a fentiekben leírt eszközök és képességek (támogatás) nélkül a hazai jogászszakma le fog maradni más országok jogászaitól. Számos, kereskedelmi forgalomban kapható terméknél tapasztalhatjuk, hogy a kisebb nyelven elérhető megoldás nem egyenértékű egy angol nyelvű megoldással, és ez a nyelvi alapú szakadék csak nőni fog, a legfontosabb „versenyszámokban" nem az adott ország adott nyelvére szabott eszközökkel versenyeznek. Ha ezeket a lemaradásokat, különösen a nyelvgenerálás terén, nem tudjuk ledolgozni, azzal közvetlenül nem is az adott ország adott nyelvészeti tudománya veszít a legtöbbet, hanem éppen a jog hétköznapi felhasználói, legyen szó akár jogászi hivatásról, akár jogkereső közönségről. Persze a lakossági felhasználásnál ettől még sokáig megmaradhat az adott kis nyelv kizárólagossága (főleg családjogban, öröklési jogban), de az üzleti területen már most is számos lehetőség van arra, hogy pusztán kényelmi okokból olyan szerződéses nyelvet (és azzal együtt akár irányadó jogot, bíróságot, választottbíróságot) válasszanak a felek, amelyet jóval több eszköz és informatikai szolgáltatás támogat (ahogy ez ma történik például Málta szigetén is).[45]

Felhasznált irodalom

Branting, L. Karl – Charles B. Callaway – Bradford W. Mott – James C. Lester: *Integrating Discourse and Domain Knowledge for Document Drafting.* Proceedings of the 7th International Conference

[44] James A. Sprowl: Automating the Legal Reasoning Process: A Computer That Uses Regulations and Statutes to Draft Legal Documents. *American Bar Foundation Research Journal,* 4. (1979), 1. 5.

[45] Köszönettel tartozom Csányi Gergely Márk NLP mérnöknek, aki a kéziratot elolvasva értékes tanácsaival segítette annak pontosabbá, szakszerűbbé tételét.

on Artificial Intelligence and Law, Association for Computing Machinery. (H. n.), (k. n.), 1999. Online: https://doi.org/10.1145/323706.323800

Branting, L. Karl: *An Issue-Oriented Approach to Judicial Document Assembly.* Proceedings of the 4th International Conference on Artificial Intelligence and Law, Association for Computing Machinery. (H. n.), (k. n.), 1993. Online: https://doi.org/10.1145/158976.159005

Carlini, Nicholas – Florian Tramer – Eric Wallace – Matthew Jagielski – Ariel Herbert-Voss – Katherine Lee – Adam Roberts – Tom Brown – Dawn Song – Ulfar Erlingsson – Alina Oprea – Colin Raffel: *Extracting Training Data from Large Language Models.* (H. n.), (k. n.), 2021. június 15. Online: https://arxiv.org/abs/2012.07805

Chalkidis, Ilias – Manos Fergadiotis – Prodromos Malakasiotis – Ion Androutsopoulos: *Neural Contract Element Extraction Revisited: Letters from Sesame Street.* (H. n.), (k. n.), 2021. február 22. Online: https://arxiv.org/abs/2101.04355

Chalkidis, Ilias – Manos Fergadiotis – Prodromos Malakasiotis – Nikolaos Aletras – Ion Androutsopoulos: *LEGAL-BERT: The Muppets Straight out of Law School.* Conference: Findings of the Association for Computational Linguistics: EMNLP 2020. (H. n.), (k. n.), 2020. október 6. 2898–2904. Online: https://doi.org/10.18653/v1/2020.findings-emnlp.261

Fedus, William – Barret Zoph – Noam Shazeer: *Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity.* (H. n.), (k. n.), 2022. június 16. Online: https://arxiv.org/abs/2101.03961

Feldmann Ádám – Hajdu Róbert – Indig Balázs – Sass Bálint – Makrai Márton – Mittelholcz Iván – Halász Dávid – Yang Zijian Győző – Váradi Tamás: *HILBERT, magyar nyelvű BERT-large modell tanítása felhő környezetben.* XVII. Magyar Számítógépes Nyelvészeti Konferencia. Szeged, Szegedi Tudományegyetem Természettudományi és Informatikai Kar, Informatikai Intézet, 2021. 29–36. Online: http://real.mtak.hu/id/eprint/120856

Ferreira, Thiago Castro – Chris van der Lee – Emiel van Miltenburg – Emiel Krahmer: *Neural Data-to-Text Generation: A Comparison between Pipeline and End-to-End Architectures.* (H. n.), (k. n.), 2019. november 27. Online: http://arxiv.org/abs/1908.09022

Indurkhya, Nitin – Fred J. Damerau (szerk.): *Handbook of Natural Language Processing.* Chapman & Hall/CRC, Boca Raton, 2010. Online: https://doi.org/10.1201/9781420085938

Jurafsky, Daniel – James H. Martin*: Speech and Language Processing.* (H. n.), (k. n.), 2020. Online: https://web.stanford.edu/~jurafsky/slp3/ed3book_dec302020.pdf

Kilián Imre: *Szemantikai alapú jogi tudás-menedzsment technológiák.* PhD-értekezés. Pécs, Pécsi Tudományegyetem Állam- és Jogtudományi Kar, Doktori Iskola, 2013. Online: https://ajk.pte.hu/files/file/doktori-iskola/kilian-imre/kilian-imre-vedes-ertekezes.pdf

Lauritsen, Marc: *Current Frontiers in Legal Drafting Systems.* A Working Paper for the 11th International Conference on AI and Law. Stanford, Stanford University, 2007. Online: https://static1.squarespace.com/static/571acb59e707ebff3074f461/t/5946f1e39de4bb69d253380c/1497821669156/CurrentFrontiers.pdf

Marković, Marko – Stevan Gostojić: *Knowledge-Based Legal Document Assembly.* (H. n.), (k. n.), 2020. szeptember 14. Online: https://doi.org/10.48550/arXiv.2009.06611

Nemeskey, Dávid Márk: *Introducing HuBERT.* XVII. Magyar Számítógépes Nyelvészeti Konferencia. Szeged, Szegedi Tudományegyetem Természettudományi és Informatikai Kar, Informatikai Intézet, 2021. Online: https://hlt.bme.hu/media/pdf/huBERT.pdf

Reiter, Ehud – Robert Dale: *Building Natural Language Generation Systems.* Cambridge, Cambridge University Press, 2000. Online: https://doi.org/10.1017/CBO9780511519857

Reiter, Ehud: Academic NLG Should Not Fixate on End-to-End Neural. *Ehud Reiter's Blog*, 2020. december 1. Online: https://ehudreiter.com/2020/12/01/dont-fixate-on-end-to-end-neural/

Reiter, Ehud: Generated Texts Must Be Accurate! *Ehud Reiter's Blog*, 2019. szeptember 26. Online: https://ehudreiter.com/2019/09/26/generated-texts-must-be-accurate/

Reiter, Ehud: NLG vs Templates: Levels of Sophistication in Generating Text. *Ehud Reiter's Blog*, 2016. december 18. Online: https://ehudreiter.com/2016/12/18/nlg-vs-templates/

Reiter, Ehud: NLG vs. Templates. *Proceedings of the Fifth European Workshop on Natural Language Generation*. Leiden, (k. n.), 1995. 95–105.

Sprowl, James A.: Automating the Legal Reasoning Process: A Computer That Uses Regulations and Statutes to Draft Legal Documents. *American Bar Foundation Research Journal*, 4. (1979), 1. 1–81. Online: https://doi.org/10.1017/CBO9780511519857

Yang Zijian Győző – Perlaki Attila – Laki László János: *Automatikus összefoglaló generálás magyar nyelvre BERT modellel*. XVI. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, 2020. 343–353. Online: http://acta.bibl.u-szeged.hu/id/eprint/67658

Zhang, Yian – Alex Warstadt – Haau-Sing Li – Samuel R. Bowman: *When Do You Need Billions of Words of Pretraining Data?* (H. n.), (k. n.), 2020. november 10. Online: https://arxiv.org/abs/2011.04946