Így írtok ti

Nem sztenderd szövegek hibatípusainak detektálása gépi tanulásos módszerrel

Dömötör Andrea[1,2] , Yang Zijian Győző[1,3]

[1] MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport,
[2] Pázmány Péter Katolikus Egyetem, Bölcsészet- és Társadalomtudományi Kar,
[3] Pázmány Péter Katolikus Egyetem, Információs Technológiai és Bionikai Kar,
e-mail:{domotor.andrea, yang.zijian.gyozo}@itk.ppke.hu

Kivonat A szövegek automatikus minőségbecslése nem csak a gépi fordítások kiértékelésénél, hanem egynyelvű szövegek esetén is fontos feladat lehet, hiszen ezek egyrészt bemenetét képezik a különböző természetesnyelvi feldolgozó rendszereknek, másrészt korpusznyelvészeti kutatások nyersanyagául is szolgálnak. A cikkben egy olyan gépi tanuláson alapuló minőségbecslő rendszert mutatunk be, amely kifejezetten egynyelvű, emberek által létrehozott szövegekhez készült. Az eredményeink a rendszer hatékonyságának mérésén kívül arról is érdekes tanulságokkal szolgálnak, hogy mik az internetes informális szövegek nyelvi jellemzői, és hogy mennyiben térnek el az emberi szövegek minőségi problémái a gépi fordítók által generált tipikus hibáktól.

Kulcsszavak: minőségbecslés, korpusznyelvészet, természetesnyelvi elemzés

1. Bevezetés

A szövegek minőségbecslésének igénye eddig elsősorban a gépi fordító rendszereknél merült fel. A fordítóprogram által adott output minőségének becsült pontszáma hasznos információ a felhasználónak, és ennek automatikus mérésére több elterjedt módszer is létezik. A minőségbecslés azonban nem csak a gépi fordítók teljesítményének értékelésével kapcsolatban lehet releváns, hanem egynyelvű, emberek által létrehozott szövegek esetén is.

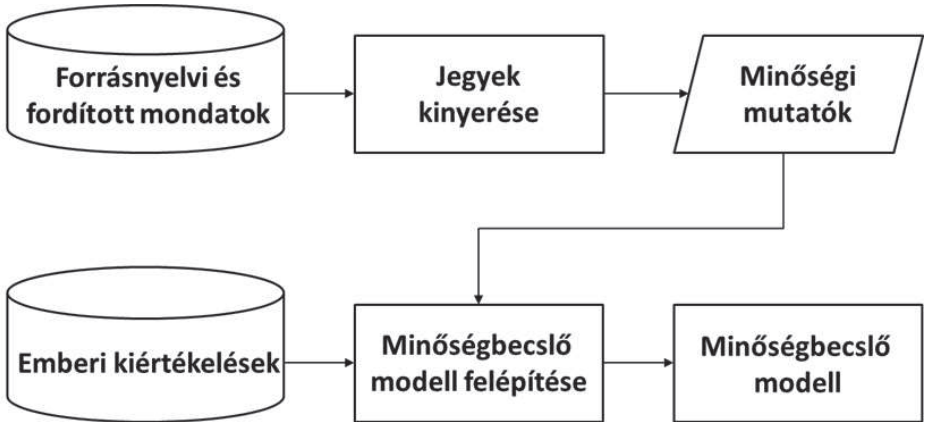
Az egyre nagyobb számban rendelkezésre álló korpuszokból nagy mennyiségben lehet hozzájutni nyersanyaghoz különböző nyelvtechnológiai, alkalmazott vagy akár elméleti nyelvészeti kutatásokhoz. A kapott szövegek azonban sok esetben nem követik a sztenderdet, és ez megnehezítheti a kutatók dolgát. Éppen ezért szükség van egy olyan eszközre, amelyik - a gépi fordítások minőségbecslő rendszereihez hasonlóan - információt tud adni az adott szövegek minőségéről és arról, hogy a feldolgozásuk előtt milyen esetleges normalizáló lépésekre lehet szükség. Ehhez nem elégséges egy meglévő, gépi fordítónak készült minőségbecslőt az adott nyelvre betanítani, hiszen az emberek által írt szövegek hibái egészen más jellegűek, mint amiket egy gépi fordító követ el. Ebben a cikkben egy olyan

gépi tanulásos rendszert mutatunk be, amely mind a tanulókorpusz, mind a jegyek tekintetében kifejezetten az egynyelvű, emberek által létrehozott szövegekre lett adaptálva.

Reményeink szerint a rendszer hasznos eszköz lesz a különböző természetesnyelv-feldolgozó alkalmazások számára azzal, hogy előzetes információt ad az input minőségéről, így például segíthet eldönteni egy automatikus mondatelemzőnek, hogy „megküzdjön"-e a mondattal, vagy jelezzen hibát. Emellett bízunk abban, hogy az egynyelvű szövegekre „szakosodott" minőségbecslő jó segédeszköz lehet a korpuszokat használó nyelvészeknek is.

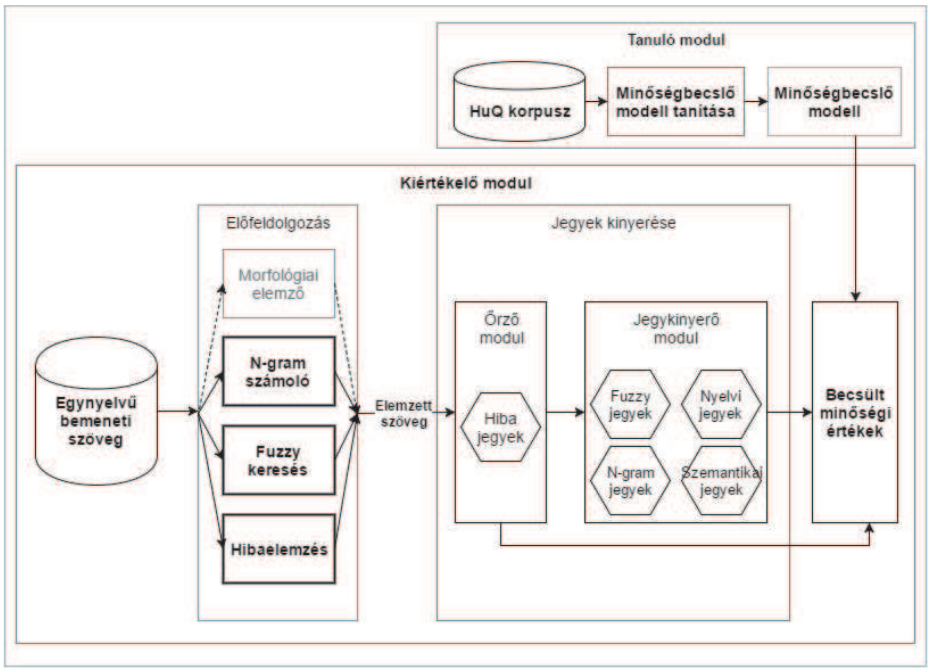
2. Kapcsolódó munkák

A hagyományos minőségbecslő módszer (lásd 1. ábra) különböző minőségi mutatókat nyer ki a forrásnyelvi és a gép által lefordított mondatokból, majd gépi tanulással betanítja azokat az emberek által kiértékelt mutatókra. Az így létrehozott modell segítségével tudja megbecsülni az új, ismeretlen mondatok minőségét. Mivel a gépi tanulás modellje emberi kiértékelésen alapszik, ezért a becsült értékek magasan korrelálnak az emberi minőségítéletekkel.



1. ábra. A minőségbecslő modell

A QuEst++ [8] rendszer szó szintű elemző része tartalmaz egynyelvű kiértékeléseket, többek között nyelvimodell-jegyeket, szintaktikai jegyeket, célnyelvi kontextusjegyeket stb. De ezek csupán egy apró részét képezik a teljes rendszernek, amely nem kifejezetten egynyelvű szövegek minőségbecslésének céljával készült.



2. ábra. A πRate rendszer

3. πRate minőségbecslő rendszer

A πRate [9] (lásd 2. ábra) egy egynyelvű minőségbecslő rendszer. A módszere a gépi fordításhoz használt minőségbecslés metodikáját (lásd 1. ábra) veszi alapul. A rendszernek két fő modulja van: a tanuló modul és a kiértékelő modul. A tanuló modul legfőbb feladata, hogy betanítja a minőségbecslő modellt. Ez a modul alapvető működésében megegyezik a hagyományos gépi fordításnál használt minőségbecslő modell tanuló moduljával. A tanításhoz emberek által kiértékelt egynyelvű korpuszt használtunk. A gépi tanuláshoz különböző nyelvi és statisztikai jegyeket vettünk fel: pl.: nyelvi jegyek; n-gram jegyek; fuzzy jegyek; hibajegyek stb. A korpuszból a jegyek segítségével kinyertük a minőségi mutatókat, majd a mutatók segítségével betanítottuk a minőségbecslő modellt az emberek által kiértékelt minőségi mutatókra.

A másik fontos modul a kiértékelő modul. Ez első lépésként beolvassa a bemeneti szöveget, majd az előfeldolgozó fázisban elemzi azt. Az így elemzett szöveget adjuk tovább a jegykinyerő (feature extraction) modulnak, amely a különböző jegyek és a betanított minőségbecslő modell segítségével előállítja a minőségi mutatókat.

A bemenet lehet nyers vagy elemzett szöveg. A πRate rendszer egyik előnye, hogy feladatorientált architektúrával rendelkezik, ezért ha a bemenet már elemzett, akkor a morfológiai elemzés műveletét kihagyhatjuk az előfeldolgozó

fázisban. Így képesek vagyunk optimalizálni az erőforrást és ezáltal a teljesítményt.

A kiértékelő modulnak három fő része van: előfeldolgozó modul, ellenőrző modul és jegykinyerő modul.

A rendszer képes inkrementálisan növekvő bemenetet kezelni. Amint a szöveg beérkezik a πRate rendszerbe, az előfeldolgozó modul morfológiailag elemzi a szöveget (ha az még nem elemzett), kiszámolja az n-gram valószínűségeket stb. Majd az elemzett szöveget továbbküldi a jegykinyerő modul számára, ahol először az ellenőrző modul ellenőrzi le azt a hibajegyek segítségével. Ha a szöveg hibamértéke meghalad egy megadott küszöbértéket, akkor az ellenőrző modul felhatalmazást kaphat megszakítani a folyamatot vagy szűrni, „cenzúrázni" a szöveget. Máskülönben továbbengedi a többi jegy számára, és a minőségbecslő modell a saját hibaértékeit használja fel minőségi mutatóként.

A jegyek kinyerése után a minőségbecslő modell a minőségi mutatók alapján kiszámolja a becsült értékeket (pl.: aktuális mondat minősége, eddig beolvasott összes szöveg globális minősége stb.).

4. Módszerek és mérések

A minőségbecslő modell felépítéséhez egynyelvű jegyekre van szükségünk, amelyek segítségével kinyerjük a minőségi mutatókat. A tanítás során a jegyek egy egynyelvű korpuszból nyerik ki a szükséges értékeket. Majd gépi tanulással emberek által kiértékelt minőségi mutatókra tanítjuk be a modellt (lásd 1. ábra). A πRate rendszer felépítéséhez JAVA EE-t használtunk.

4.1. Egynyelvű korpusz

Az egynyelvű, emberek által létrehozott szövegek hibái más jellegűek, mint azok, amelyeket egy gépi fordító követ el, ezért az egynyelvű szövegek minőségbecslése másfajta tanítóanyagot és jegykészletet igényel. A tanító- és tesztkorpuszt az MNSz2-ből [5] hoztuk létre. Ehhez a beszélt nyelvi és a személyes alkorpuszokból kértünk le random adatot. Azért választottuk ezeket a szövegtípusokat, mert feltételezhetően ezekben a leggyakoribb a sztenderdtől való eltérés. A tanítókorpusz annotálása nyelvészeti alapokon, kézzel történt. Kétféle annotációt használtunk: Likert-skálát (1-5) és osztályozási modellt. A Likert-pontszámok esetén nem a szubjektív emberi értékelést vettük figyelembe, hanem azt próbáltuk pontozni, hogy a mondat elemzése várhatóan mennyire okozhat nehézséget (a normától való eltérésből adódóan) egy szabály alapú gépi eszköz számára. A pontszámokat a helyesen elemezhető összetevők és összetevős szerkezetek arányából számoltuk ki. Az elemzendő összetevők alatt az NP-ket és névutós szerkezeteket, az igék, igekötők és vonzatok kapcsolatát, illetve a tagmondatokat és végül a teljes mondatot értjük. Az (1a)-ban és (1b)-ben látható példákban []-lel jelöljük a felismerhető, és []!-lel a nem felismerhető összetevőket.

(1) a. *[Emberünk ugyanis [állateledel és kisállat kereskedést [tart fenn]]! .]!*

b. *[Ugyanis [törvénysértést [követett el]] [az erkölcsrendész ismerosöm szerint]!]!*

Az osztályozás során a hibátlan mondatok kaptak 5 pontot, a 20%-nál kevesebb felismerhetetlen összetevőt tartalmazók 4-et, a 20 és 39% közötti hibaarányúak 3-at, a 40 és 59% közöttiek 2-t, a legrosszabb 1-es pontszámot pedig azok a mondatok kapták, ahol az összetevők legalább 60%-a minősült elemezhetetlennek. Ez az annotálási rendszer bonyolultnak és indokolatlanul időigényesnek tűnhet, de például (1a)-ban láthatjuk, hogy az *állateledel- és kisállatkereskedés* rossz helyesírásán az emberi értelmezés ugyan könnyen túllendül, egy számítógépes elemző számára azonban gyakorlatilag értelmezhetetlen (vagy félreérthető) így a mondat. Az értékelés tehát azért nem pusztán emberi ítélettel történt, mert a pontszámokkal elsősorban gépi eszközöket szeretnénk informálni, így a pontozásnál ki kellett iktatni az ember „természetes normalizáló képességét".

Az így kapott érték információt adhat az elemző rendszernek az input megbízhatóságáról. Ugyanakkor a hiba típusa még sokkal informatívabb lehet egy gépi eszköz számára: ha a minőségbecslő megbízhatóan detektálja a hiba jellegét, az eszköz alkalmazni tudja a megfelelő normalizáló modult (például: helyesírásellenőrző, ékezet-visszaállító). Az osztályozási modellünk öt hibatípust tartalmaz, amelyek a beszélt nyelvi és az informális internetes szövegekre jellemzők. Ezek a következők:

1. Központozás hibái (hiánya), nagybetűk elhagyása
2. Elírások, helyesírási és nyelvi hibák
3. Idegen nyelvű, idegen szavakat tartalmazó szövegek
4. Ékezetek hiánya
5. Nehezen elemezhető beszélt nyelvi vagy informális szövegek (ismétlések, elakadások, szleng, rövidítések, emotikonok stb.)

A fenti osztályokon kívül meghatároztunk még egy szegmentálásihiba-osztályt is, ide azokat az eseteket soroltuk, amikor a korpuszból kapott adat valójában nem volt mondat, vagy nem egy mondat volt. Továbbá a hibaosztályok mellett természetesen szerepelt egy osztály a hibátlan mondatok számára is.

Ha egy mondat több hibatípusba is besorolható, az egycímkés tanításhoz kiválasztottunk egy fő hibát, azt, amelyik a szöveg nagyobb részét lefedi. Például (2a)-ban elírás is szerepel (*gyeppet*), de jobban jellemzi a szöveget az ékezetek hiánya. (2b)-ben pedig, bár egy mondatközi írásjel is hiányzik, de jelentősebb a helyesírási hibák előfordulása, hiszen a mondat nyolc szavából három is hibás.

(2) a. *De gyeppet sem siettek el a dolgot, es mindharman a jarda kozepen tanyaztak.*
Ékezetek hiánya

b. *Valamelyik ujságban olvastam hogy állitolag fel akarják újitani.*
Helyesírási hibák

A többcímkés osztályozási modellhez minden mondat 3 címkét kapott, ennél több hibaosztályba egyik adat sem volt besorolható. Ahol háromnál kevesebb hibatípus fordult elő, ott a hiányzó helyekre a „helyes" címkét kapták a mondatok.

4.2. Egynyelvű jegyek

A minőségbecslő modellünk 36 különböző típusú jegyet használ, amelyeket jellegük alapján az alábbi kategóriákba soroltuk:

- nyelvi jegyek:
  - főnevek, igék, igekötők, melléknevek, határozószók, kötőszók, névmások, névelők, indulatszók aránya a mondatban;
  - főnevek és igék aránya a mondatban;
  - főnevek és melléknevek aránya a mondatban;
  - igék és igekötők aránya a mondatban;
  - főnevek és névelők aránya a mondatban;
  - tokenek száma;
  - átlagos szóhossz a mondatban;
- n-gram jegyek:
  - a mondat nyelvmodell valószínűsége;
  - a mondat nyelvmodell perplexitása;
  - a mondat szótöveinek, szófaji címkéinek nyelvmodell valószínűsége;
  - a mondat szótöveinek, szófaji címkéinek nyelvmodell perplexitása;
- neurális nyelvmodell jegyek:
  - 1-gram, 2-gram és 3-gram perplexitás;
- hibajegyek:
  - ismeretlen szavak aránya a mondatban;
  - ékezetes szavak aránya a mondatban;
  - írásjelek aránya a mondatban.

Az n-gram modell felépítéséhez (az n-gram jegyekhez) szintén az MNSz2 egy részkorpuszát használtuk, amely 98500 lemmatizált és elemzett mondatot tartalmaz.

A neurális nyelvmodell tanításához a Pázmány Korpuszból [2] vettünk 1 millió mondatot. A nyelvmodell felépítéséhez egy GRU-alapú (Gated recurrent unit) RNN architektúrát használtunk, amely 6 epochig tanult. Továbbá a modellünkhöz szóbeágyazást (word embedding) is alkalmaztunk, amelyhez a [4] által készített magyar nyelvű szóbeágyazási modellt használtuk fel.

4.3. Mérések

A kutatásunk során kétféle mérést végeztünk: osztályozást és regressziót. A Likert értékek segítségével regressziós modelleket építettünk, a hibaosztályok segítéségével pedig osztályozási modelleket készítettünk. Az osztályozás esetében végeztünk egycímkés (a fő hibaosztállyal) és többcímkés osztályozást is. A regresszióhoz és az egycímkés osztályozáshoz a WEKA [3] szoftvert, míg a többcímkés osztályozáshoz a MEKA [7] rendszert használtuk.

Többféle tanuló algoritmust is kipróbáltunk, melyek közül az egycímkés osztályozáshoz a szupport vektor gép, a regresszióhoz a szupport vektor regresszió, a többcímkés osztályozáshoz pedig a véletlen erdő (random forest) alapú „Classifier Chains" [6] módszer érte el a legjobb eredményt, ezért az eredmények fejezetben csak az ezekkel a módszerekkel betanított modellek eredményeit mutatjuk. Az annotált tanítóanyag segítségével az alábbi három minőségbecslő modellt építettük:

- LS modell: regressziós minőségbecslő modell a Likert értékeket felhasználva. A Likert értékek 1-től 5-ig terjedő egész számok.
- CS modell: egycímkés osztályozási minőségbecslő modell a fő hibaosztályokat felhasználva. Összesen 6 hibaosztály és a helyes mondat osztályozási címkéje.
- CCS modell: többcímkés minőségbecslő modell az osztályzási értékeket felhasználva. Minden mondathoz 3 db osztályt rendeltünk, az első osztály a főcímke, ami vagy a fő hibaosztály, vagy a helyes mondat címkéje. A második és a harmadik osztály a mellékhibá(ka)t tartalmazza, ha nincs ilyen, a helyes mondat címkéjét viselik.

Továbbá végeztünk optimalizációt is. A gépi fordítás kiértékelésének optimalizációja alapján [1] ha kivesszük a kevésbé releváns jegyeket a rendszerből, kevesebb jeggyel magasabb minőséget tudunk elérni. A „forward selection" [10] módszerével az alábbi optimalizált modelleket hoztuk létre:

- OptLS modell: optimalizált LS modell.
- OptCS modell: optimalizált CS modell.

5. Eredmények és kiértékelések

5.1. A πRate rendszer

A πRate rendszer kiértékeléséhez az MAE (mean absolute error - átlagos abszolút eltérés), az RMSE (root mean square error - átlagos négyzetes eltérés gyöke), a Pearson-féle korreláció, a helyesen osztályozott egyed (Correctly Classified Instances - CCI), a Hamming-veszteség (Hamming loss) és a pontosság (accuracy) mértékeket használtuk. A teszteléshez minden esetben tízszeres keresztvalidálást használtunk. Az 1. és 2. táblázatokban látható, hogy a 36-os jegykészlet ∼77,1%-os korrelációt és ∼64,48% helyesen osztályozott egyedet ért el.

Az optimalizálást a „forward selection" módszerrel végeztük. Az optimalizálás utáni eredményeket a 1. és 2. táblázatok második sorai tartalmazzák.

- Az OptLS jegykészlet, amelyik 15 jegyet használ, nagyjából ugyanolyan korrelációt ért el, mint a teljes jegykészlet, de lényegesen kevesebb munkával.
- Az OptCS jegykészlet, amelyik 28 jegyet használ, nagyjából ugyanolyan eredményt ért el, mint a teljes jegykészlet, de kevesebb munkával.

Amint látható, a Likert-skála modell megfelelő korrelációval működik, ám az egycímkés osztályozási modell kevesebb eredményességet mutat. Ennek magyarázata részben az annotálási módszer is lehet. Amint 4.1-ben említettük, egy

	Korreláció	MAE	RMSE
LS modell - 36 jegy	0.7712	0.7121	1.0047
OptLS készlet - 15 jegy	0.7777	0.7226	0.9625

1. táblázat. Az LS modell és az OptLS jegykészlet értékelése

	CCI	MAE	RMSE
CS modell - 36 jegy	64.48%	0.214	0.3171
OptCS készlet - 28 jegy	65.17%	0.2137	0.3167

2. táblázat. A CS modell és az OptCS jegykészlet értékelése

mondat csak egy hibaosztályba tartozhat az annotáció szerint, holott a valóságban többféle hibát is tartalmazhat. Azaz, az osztályozó feladata valójában az, hogy meghatározza az elsődleges hibatípust, ami igencsak nehéz is lehet, ha a mondatban egyéb, kevésbé releváns hibák is megtalálhatók.

A sikertelenség okainak pontosabb feltérképezésére elkészítettük a méréshez tartozó tévesztési mátrixot (3. táblázat). Ebből látható, hogy a rendszer jól boldogul az ékezet nélküli és a hibátlan szövegek besorolásával, a többi osztálynál viszont gyengébb eredményeket mutat. Különösen figyelemre méltó, hogy az idegen nyelvű és a beszélt nyelvű szövegekre egyszer sem tippelt a program, úgy látszik, ezek jellemezhetők a legkevésbé a rendszer által használt jegyekkel.

	Ékezet	Hí.	Id. nyelv	Beszélt nyelv	Központozás	Szegm.	Helyes	F-score
Ékezet	107	0	0	0	1	0	7	0.915
Helyesírás	2	30	0	0	40	1	72	0.267
Idegen nyelvű	1	0	0	0	1	3	3	0.0
Beszélt nyelvi	0	6	0	0	28	0	19	0.0
Központozás	3	23	0	0	127	1	83	0.547
Szegmentálás	5	10	0	0	14	12	21	0.304
Helyes	1	3	0	0	16	0	385	0.774

3. táblázat. Tévesztési mátrix

5.2. Az optimalizált jegykészlet

A 4. és 5. táblázatokban láthatók az optimalizált jegykészletek, a jegyek relevanciája szerint rendezve.

A Likert-skála szerinti minőségbecslés szempontjából legrelevánsabb nyelvi és hibajegyek az ékezetekkel, a tokenszámmal és az írásjelekkel függnek össze. Ez az eredmény nyelvészeti szempontból nézve nem meglepő. Az informális írásbeli kommunikációban (azaz legjellemzőbben az internetes szövegekben) az ékezetek és írásjelek elhagyása a legtipikusabb sztenderdtől való eltérés. Utóbbi a gépi

Jegy
Ékezetes karakterek száma
Ékezetes karaktert tartalmazó szavak / összes szó
A tokenek n-gram perplexitása (ismeretlen szavakkal együtt)
A szövegben előforduló elemzési címkék n-gram perplexitása (ismeretlen szavakkal együtt)
1-gram perplexitás (neurális nyelvmodell)
A szövegben előforduló elemzési címkék n-gram perplexitása (ismeretlen szavak nélkül)
Ismeretlen szavak száma
A szótövek n-gram perplexitása (ismeretlen szavak nélkül)
A szövegben előforduló szófajcímkék n-gram perplexitása
A szótövek n-gram perplexitása (ismeretlen szavakkal együtt)
Indulatszók aránya
Mondatközi írásjelek száma / mondatvégi írásjelek száma
Tokenek száma
Főnevek száma / igék száma
A szófajcímkék n-gram perplexitása (ismeretlen szavakkal együtt)

4. táblázat. 15 jegyre optimalizált jegykészlet a Likert-modellhez

feldolgozásban szegmentálási problémákat okozhat, ezzel magyarázható a sokszor extrém magas tokenszám. Ha tehát egy mondat (vagy amit az elemző egy mondatnak hisz) nagyon hosszú, a minőségbecslő rendszer gyanakodhat, hogy egy szerkesztetlen szövegről van szó, ami gyakran együtt jár az alacsony minőséggel. Az optimalizált jegykészletben előfordul még az ismeretlen szavak és az indulatszók aránya is. Ezen címkék magas száma szintén az informális szövegek sajátossága.

Az egycímkés osztályozási modell optimalizált jegyei között már több olyan jegy is megjelenik, amely valódi nyelvi hibára utalhat, ilyenek például a főnevek és névelők, az igék és igekötők aránya vagy a névmások száma. Ezek szükségesek ahhoz, hogy a modell a nyelvtani helyességre (helytelenségre) vonatkozó hibaosztályokat is detektálni tudja. Az ilyen típusú jegyek azonban kevésbé tűnnek relevánsnak a Likert-modell esetén, ami azt mutatja, hogy az egynyelvű korpuszokból származó szövegek minőségi problémái nagyrészt nem nyelvi természetűek a szó szoros értelmében véve.

5.3. A hibatípusok összefüggései

A 6. táblázat az egyes hibaosztályok átlagos Likert-pontszámát mutatja. Eszerint a normalizálást tekintve az ékezet-visszaállítás, a nyelvfelismerés és a megfelelő mondatokra és tagmondatokra bontás (praktikusan írásjel-visszaállítás) jelentheti a legnagyobb segítséget egy gépi feldolgozó eszköz számára.

A hibatípusok összefüggéseit főkomponens analízissel vizsgáltuk meg. A kapott eredményeket a 7. táblázat tartalmazza. Ebből az derül ki, hogy a vizsgált

Jegy
Ékezetes karaktert tartalmazó szavak / összes szó
Írásjelek aránya
Ékezetes karakterek száma
Névmások száma
A mondat szótöveinek n-gram valószínűsége
Igék aránya
Főnevek száma / névelők száma
Főnevek száma / igék száma
Tokenek száma
A szövegben előforduló elemzési címkék n-gram valószínűsége
Kötőszavak aránya
írásjelek aránya
Határozószók aránya
A szófajcímkék n-gram valószínűsége
A tokenek n-gram perplexitása (ismeretlen szavakkal együtt)
Átlagos szóhossz a mondatban
A mondat szótöveinek n-gram perplexitása (ismeretlen szavakkal együtt)
Melléknevek aránya
A tokenek n-gram valószínűsége
A szófajcímkék perplexitása (ismeretlen szavakkal együtt)
2-gram perplexitás (neurális nyelvmodell)
Számnevek aránya
Igék száma / igekötők száma
1-gram perplexitás (neurális nyelvmodell)
3-gram perplexitás (neurális nyelvmodell)
Főnevek száma / melléknevek száma
A mondat szótöveinek n-gram perplexitása (ismeretlen szavak nélkül)
Determinánsok aránya

5. táblázat. 28 jegyre optimalizált jegykészlet az egycímkés osztályozási modellhez

	Átlagos pontszám
Ékezetek hiánya	1.16
Idegen nyelvű szövegek	1.63
Szegmentálási hibák	1.74
Elütések, helyesírási és nyelvi hibák	2.69
Írásjelek hibái (hiánya), nagybetűk elhagyása	3.20
Nehezen elemezhető beszélt nyelvi szövegek	3.28

6. táblázat. A hibaosztályok átlagos Likert-pontszáma

informális műfajokban a kevésbé gondos szövegalkotás több jelenségben is megnyilvánulhat egyszerre. Így például számíthatunk arra, hogy az ékezeteket nem tartalmazó szöveg jó eséllyel nyelvi szempontból sem fog megfelelni a sztenderdnek (l. 1. faktor). A szegmentálási hibák az elemzés szerint gyakran az idegen nyelvű szövegekkel járnak együtt (l. 2. faktor), az írásjelek elhagyása viszont úgy tűnik, hogy a többi szempontból kifogástalan szövegekre is nagy arányban jellemző (3. faktor). Mivel az egycímkés osztályozási modell tanuló korpusza jelenleg csak egy (elsődlegesnek tekintett) hibatípust rendel a mondatokhoz, fontos ezeknek az összefüggéseknek az ismerete.

1. faktor	beszélt nyelvi szöveg	nyelvi hibák	ékezetek hiánya
2. faktor	szegmentálási hibák	idegen nyelvű szövegek	
3. faktor	központozási hibák		

7. táblázat. A hibatípusok összefüggései főkomponens analízissel

5.4. Többcímkés osztályozási modell

A többcímkés osztályozási modell (8. táblázat) a fő hibaosztály detektálásában hasonló eredményességet mutat, mint az egycímkés modell. A második és harmadik címke pontosságának javulása annak is köszönhető, hogy ezek egyre nagyobb arányban tartoztak a hibátlan osztályba. Mindemellett az 56.6%-os pontos egyezés, a feladat komplexitását és a tanuló adatok kis számát tekintve, viszonylag jó eredménynek mondható.

	Fő hibaosztály	2. hibaosztály	3. hibaosztály
Pontosság címkénként	0.652	0.835	0.964
Pontos egyezés		0.566	
Hamming veszteség		0.183	

8. táblázat. A többcímkés osztályozási modell eredményei

6. Összegzés

A cikkben egy olyan egynyelvű szövegek minőségbecslésére tervezett rendszer felépítését és működését mutattuk be, amely jól alkalmazható lehet a korpusznyelvészetben vagy a természetesnyelvi elemző rendszerek előfeldolgozó moduljában. Az eredmények azt mutatták, hogy az emberek által létrehozott egynyelvű szövegek, a gépi fordítók által generáltakkal ellentétben, nagyobbrészt nem nyelvtani

hibákat tartalmaznak. Ezen szövegek minőségi problémái sokkal inkább az internetezők írási szokásaiból adódnak, mint például az ékezetek vagy az írásjelek elhagyása.

A rendszer értékeléséből az derült ki, hogy a hibatípusok detektálása még fejlesztésre szorul, egyelőre csak a hibátlan és az ékezet nélküli szövegek elkülönítésében működik megbízhatóan. Az általános minőségi mutató (Likert-pontszám) megbecsülésében azonban jó eredményt értünk el.

További általános észrevétel még, hogy a jegykészlet-optimalizáció nagy jelentőséggel bír. Az eredményeink szerint a jegykészlet csökkentésével javítható a teljesítmény, és az erőforrás-felhasználás is kevesebb lesz. Az egynyelvű szövegek optimális jegykészlete eltért attól, ami a korábbi kutatásokban a fordítások esetén tapasztalható volt, elmondható tehát, hogy az optimális jegykészlet feladatfüggő.

Hivatkozások

1. Beck, D., Shah, K., Cohn, T., Specia, L.: Shef-lite: When less is more for translation quality estimation. In: Proceedings of the Workshop on Machine Translation (WMT) (2013)
2. Endrédy, I., Prószéky, G.: A pázmány korpusz. Nyelvtudományi Közlemények (112), 191–206 (2016)
3. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: An update. SIGKDD Explor. Newsl. 11(1), 10–18 (Nov 2009)
4. Novák, A., Novák, B.: Magyar szóbeágyazási modellek kézi kiértékelése. In: XIV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2018). Szegedi Tudományegyetem, Szeged, Hungary (2018)
5. Oravecz, C., Váradi, T., Sass, B.: The Hungarian Gigaword Corpus. In: Calzolari, N., et al. (eds.) Proceedings of the 9th International Conference on Language Resources and Evaluation. ELRA, Reykjavik, Iceland (may 2014)
6. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Mach. Learn. 85(3), 333–359 (Dec 2011), http://dx.doi.org/10.1007/s10994-011-5256-5
7. Read, J., Reutemann, P., Pfahringer, B., Holmes, G.: MEKA: A multi-label/multi-target extension to Weka. Journal of Machine Learning Research 17(21), 1–5 (2016), http://jmlr.org/papers/v17/12-164.html
8. Specia, L., Paetzold, G., Scarton, C.: Multi-level translation quality prediction with quest++. In: ACL-IJCNLP 2015 System Demonstrations. pp. 115–120. Beijing, China (2015)
9. Yang, Z.G., Laki, L.J.: Minőségbecslő rendszer egynyelvű természetes nyelvi elemzőhöz. In: XIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 37–49. Szegedi Tudományegyetem, Informatikai Tanszékcsoport, Szeged, Hungary (2017)
10. Yang, Z.G., Laki, L.J., Siklósi, B.: Quality estimation for english-hungarian with optimized semantic features. In: Computational Linguistics and Intelligent Text Processing. Konya, Turkey (2016)