PurePos 2.0: egy hibrid morfológiai egyértelműsítő rendszer

Orosz György, Novák Attila

MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport,
Pázmány Péter Katolikus Egyetem, Információs Technológiai és Bionikai Kar
1083, Budapest Práter utca 50/a
e-mail: {oroszgy, novak.attila}@itk.ppke.hu

1. Bevezetés

A szófaji egyértelműsítés és a lemmatizálás jól ismert problémái a nyelvtechnológiának. A fenti feladatokat gyakran különálló komponensek végzik egy szövegfeldolgozási láncban, ami forrása lehet a lánc teljesítményének romlásának. Az utóbbi évtizedben számos olyan eszköz jött létre, mely képes magyar nyelvű szövegek szófaji vagy morfológiai egyértelműsítésére, ilyenek pl.: a Hunpos [2], az OpenNLP[1], a `magyarlanc` [5] és a PurePos [4]. Ezek közül csak néhány [5,4] képes teljes morfológiai elemzések közti egyértelműsítésre, továbbá egy olyan hibrid láncban, ahol szabályalapú modulok is fontos szerepet töltenek be, ezek egyike sem tud maradéktalanul együttműködni társaival. A szófaji és morfológiai egyértelműsítés napjaink egyik aktuális problémája a doménadaptáció kérdése: hogyan alkalmazható egy általános nyelvi modell egy új doménen?

Írásunkban ismertetjük a PurePos rendszer továbbfejlesztett változatát, melyben az eredeti algoritmus több ponton megváltoztattuk, úgy, hogy az egy hibrid elemző lánc hasznos tagja legyen. Továbbá az eszközt olyan jellemzőkkel láttuk el, melyek lehetővé teszik használatát azon doménadaptációs feladatokban is, amikor szabályok alkalmazásával növelni lehet az elemzőlánc teljesítményét. Cikkünk végén ismertetjük a módosított rendszer megnövekedett teljesítményét.

2. A továbbfejlesztett algoritmus

A PurePos alapja olyan rejtett Markov-modellezésen alapuló algoritmusok, melyeket már számos alkalommal sikerrel használtak szófaji egyértelműsítő rendszerekben (pl. HunPos[2], TnT[1]). A tagger a címkézéshez egy lexikális és kontextuális modellt használ, melyek együttese (1) formalizálja az egyértelműsítés feladatát[2].

$$\arg\max_T P(T|W) = \arg\max_T P(W|T)P(T) \tag{1}$$

[1] http://opennlp.apache.org/

[2] T címkesorozatot, t címkét, W mondatot, míg w egy szót jelöl.

$$P(t_k|t_{k-1,k-n}) = \sum_{i=1}^{n} \lambda_i \hat{P}(t_k|t_{k-1,k-i}) \tag{2}$$

$$P(w_k|t_{k,k-m+1}) = \sum_{i=1}^{m} \lambda_i \hat{P}(w_k|t_{k,k-i+1}) \tag{3}$$

Az egyértelműsítő a lexikai és kontextuális modellek becslésére simított n-gram modelleket használ ((2) és (3)), melyek paraméterei változtathatóak, de alapesetben $n = 2$ és $m = 2$. Az interpoláció környezetfüggő módon deleted interpolation módszert használva történik (l. [1]). A PurePos a tanítóanyagban nem látott (OOV) szavak taggeléséhez az integrált morfológiai elemző analízisein túl egy szóvég alapú javasló rendszert is tartalmaz, melynek működése a [1]-ben részletezett hasonló moduljára épül. Az eredeti egyértelműsítő a lemmatizáláshoz egy maximum likelihood becslésen alapuló unigram modellt használ, míg a dekódolást a Viterbi algoritmus végezte.

2.1. A morfológiai tudás fejlettebb használata

A PurePos korábbi verziói is használtak morfológiai elemzőt az ismeretlen szavak jobb taggelése céljából, viszont nem voltak képesek teljesen kihasználni ezt az értékes tudást. Az egyik ilyen eset, amikor egy ismeretlen szóhoz az egyetlen morfológiai elemzés olyan, hogy annak címkéje a tanítóanyagban még nem fordult elő. Ekkor a tag valószínűsége 0 volt, ami vagy a jó megoldást kizárásával járt, vagy pedig azt eredményezte, hogy a tagger – a logaritmikus reprezentáció tulajdonságai miatt – a mondat elemzéssorozataihoz egyforma valószínűséget rendelt. Ezt a hibát úgy javítottuk, hogy valószínűségi értékként 1-et használunk.

Egy ettől összetettebb jelenség, amikor a helyes elemzés továbbra is ismeretlen a statisztikai rendszer számára, viszont mellette több más lehetséges annotáció is feltűnik, amikhez a trigram modell már képes gyakorisági értékeket rendelni. Ilyenkor is számtalan esetben a 0 gyakorisági érték miatt elveszett a helyes elemzés, amit az új PurePosban címkék megfeleltetésével küszöböltünk ki. A módszer alapja, hogy a tagger indításakor egy konfigurációs fájl használatával lehetősége van a címkék megfeleltetésére, ami az egyértelműsítés folyamán azt eredményezi, hogy az így megadott, tanítóanyagban nem látott elemzésekhez is a leképezett annotáció gyakorisági értékei számolódnak.

2.2. Fejlettebb szótövezés

$$\arg\max_{l} P(l|t, w) \tag{4}$$

Egyes w szavak t címkéjéhez tartozó l optimális lemmát (4) segítségével határozzuk meg. Ennek becslésére a szoftver korábbi változata a tanítóanyag alapján számolt maximum likelihood becsléssel végezte a szótövek rangsorolását. Jelen

munkánkban módosítjuk ezt az eljárást, hogy az ismeretlen szavakhoz használt guesser valószínűségi becsléseit is figyelembe vegye a szoftver.

$$P(l|t,w) = \frac{P(l,t|w)}{P(t|w)} \tag{5}$$

Ehhez (4) átírásából kapjuk (5)-öt, amiből a nevező konstans volta miatt elhagyható. A számláló eloszlására, mint korábbi munkánkban azt megmutattuk, jól alkalmazható a suffix guesser interpolált modellje. Hogy mind az unigram valószínűségek, mind pedig az utóbbi erősségeit alkalmazhassa az egyértelműsítő, ezek log-lineárisan interpolált kombinációját számoljuk (6).

$$P(l|w,t) = P(l)^{\lambda_1} P(l,t|w)^{\lambda_2} \tag{6}$$

Algoritmus 1 Az interpolált modell paramétereinek számítása

```
 1: for all (w, t, l) do
 2:     candidates ← generateLemmaCandidates(w, t)
 3:     maxUnigramProb ← getMaxProb(candidates, w, t, unigramModel)
 4:     maxSuffixProb ← getMaxProb(candidates, w, t, suffixModel)
 5:     actUnigramProb ← getProb(w, t, l, unigramModel)
 6:     actSuffixProb ← getProb(w, t, l, suffixModel)
 7:     unigramProbDistance ← maxUnigramProb − actUnigramProb
 8:     suffixProbDistance ← maxSuffixProb − actSuffixProb
 9:     if unigramProbDistance > suffixProbDistance then
10:         λ2 ← λ2 + unigramProbDistance − suffixProbDistance
11:     else
12:         λ1 ← λ1 + suffixProbDistance − unigramProbDistance
13:     end if
14:     normalize(λ1, λ2)
15: end for
```

Az interpoláció paraméterinek kalkulálásához Brants [1] ötletét használjuk, miszerint a tanítóanyagon jobban teljesítő modell nagyobb súlyt kap (vö. 1. algoritmus). Ehhez az egyes komponensek tanítása után, a korpusz összes szavára kiértékeljük a szótövező modulokat (3-8. sor), és negatív súlyokat adunk a rosszabbul teljesítőnek (9-13. sor). A tanítás végén a $\lambda_{1,2}$ paraméterek értékei normalizálásra kerülnek.

2.3. *k*-legjobb kimenet

Számos esetben igény van a tagger kimenetén a legjobbnak vélt elemzési szekvencián túl a lehetséges annotációk egy halmazára is. Ennek érdekében a PurePos a Viterbi algoritmus használatán túl támogatja a Beam-search dekódolást is, mely paraméterei futtatási opciókként állíthatóak. Az eszköz az egyes szekvenciákhoz nyilvántartja még azok rangsorolásához használt logaritmikus

valószínűségi értékeket is, amik szintén megjelenhetnek az outputon. Ezek a (7) alapján számolódnak.

$$Score(w_{1,m}, t_{1,m}) = log \prod_{i=1}^{m} P(w_k|t_{k,k-m+1})P(t_k|t_{k-1,k-n}) \tag{7}$$

2.4. Hibrid komponensek használata

A morfológiai elemző használatán túl, a rendszer lehetővé teszi még a felhasználó számára hogy további nyelvi tudással segítse a taggelés eredményességét. Így a PurePos inputján az egyes tokenekhez lehetséges elemzések és azokhoz tartozó valószínűségek is megadhatóak. Ez a képessége jól használható pl. olyan doménadaptációs feladatok esetén, amikor a céldomén egyes, különleges módon használt szavai jól körülhatárolhatóak. Ezeken túl az adaptálható input és egy morfológiai elemző használatának segítségével további, nagyobb hatótávolságú szabályok is megfogalmazhatóak. A k-legjobb elemzési opciót használva az elemző lánc építőjének további lehetősége nyílik a teljesítmény további javítására, vagy ún. self-training használatára is.

3. Eredmények

A fejezetben bemutatunk egy olyan esetet, amikor a PurePos 2.0 fent részletezett tulajdonságai használatával jelentősen sikerült javítani az elemzőlánc teljesítményén. Az Ó- és Középmagyar Korpusz [3] morfológiai annotációjának készítése során 200 dokumentum mintegy 75000 tokenjéhez kellett egyértelműsített morfológiai elemzést rendelni. Munkánk során a korpusz 80%-át tanításra használtuk, míg 10-10%-ot az algoritmus paramétereinek beállítására, illetve annak kiértékelésére.

1. táblázat. Az egyértelműsítő pontossága a teszthalmazon

	Szófaji címkézés	Teljes egyértelműsítés
PurePos 1.0	91,09%	51,32%
PurePos 2.0	96,72%	96,48%
Címkeleképezésekkel	96,75%	96,51%
Előfeldolgozó szabályokkal	96,86%	96,66%
A teljes lánc	96,89%	96,67%

A 1. táblázatban bemutatjuk a PurePos első verziójának teljesítményét, ezen túl ismertetjük még az új komponensek használatával elért teljesítményjavulást is. A bemutatott szótövező algoritmus használatával jelentős mértékben sikerült csökkenteni a hibák számát, míg a többi modul is javított a pontosságon. A címkék megfeleltetéséhez mindössze egy szabályt alkalmaztunk, mely igekötős igék eloszlását köti az igekötő nélküliekhez. A hibrid komponensben használt

szabályok mindössze két megfigyelés formalizálásával történtek. Ezek közül az egyik a mondat eleji *a* szó névelő voltát határozza meg, míg a másik gyakori foglalkozást jelentő tulajdonnevek elemzéseit egyértelműsíti. Végül fontos még megemlíteni a *k*-legjobb elemzési szekvencia használatát. Ezzel az opcióval a bemutatott legjobb teljesítményű konfiguráció hibái további csökkenthetőek akár 98,65% teljes egyértelműsítési pontosságot megközelítve.

4. Összefoglalás

Munkánkban bemutattuk a nagy pontosságú PurePos rendszer egy továbbfejlesztett változatát, mely a jobb szótövezési teljesítményen túl immár hasznos eleme lehet egy hibrid elemző láncnak is. A tagger jól használható olyan környezetekben, ahol egyszerű szabályok bevezetésével lehetséges a teljesítmény javítása. Dolgozatunkban egy használati eseten keresztül megmutattuk, hogy akár kis méretű tanítóanyag esetén is nagy pontosságú morfológiai egyértelműsítő hozható létre.

Az alkalmazás JAVA nyelven íródott, nyílt forráskódú[3] és Python nyelvhez is tartalmaz illesztést. A részletezett tulajdonságok és a megengedő felhasználási feltételek miatt is a PurePos megfelelő választás lehet elemzési feladatok egyértelműsítő komponensének.

Köszönetnyilvánítás

Ez a projekt a TÁMOP–4.2.1./B–11/2-KMR-2011-0002 és a TÁMOP–4.2.2./B–10/1-2010-0014. támogatásával készült.

Hivatkozások

1. Brants, T.: TnT - A Statistical Part-of-Speech Tagger. In: Proceedings of the sixth conference on Applied Natural Language Processing. pp. 224–231. Universität des Saarlandes, Computational Linguistics, Association for Computational Linguistics (2000)
2. Halácsy, P., Kornai, A., Oravecz, C.: HunPos: an open source trigram tagger. In: Proceedings of the 45th Annual Meeting of the ACL. pp. 209–212. Prague, Czech Republic (2007)
3. Novák, A., Orosz, G., Wenszky, N.: Morphological annotation of Old and Middle Hungarian corpora. In: Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities. pp. 43–48. Sofia, Bulgaria (2013)
4. Orosz, G., Novák, A.: PurePos – an open source morphological disambiguator. In: Proceedings of the 9th International Workshop on Natural Language Processing and Cognitive Science. pp. 53–63. Wroclaw (2012)
5. Zsibrita, J., Vincze, V., Farkas, R.: magyarlanc: A Toolkit for Morphological and Dependency Parsing of Hungarian. In: Proceedings of Recent Advances in Natural Language Processing 2013. Association for Computational Linguistics (2013)

[3] http://nlpg.itk.ppke.hu/software/purepos