

# Főnévi csoportok azonosítása szabályalapú és hibrid módszerekkel

Recski Gábor

MTA SZTAKI

Nyelvtechnológiai Kutatócsoport

e-mail: recski@sztaki.hu

## 1. Bevezetés

Cikkünkben először egy a magyar főnévi csoportok azonosítására épített mondattani elemzőt (parser-t) mutatunk be, melynek pontossága megközelíti a hasonló célú, gépi tanuláson alapuló rendszerünk (**hunchunk**, Recski et al. 2009) eredményeit, majd megmutatjuk, hogy a két eszköz egyesítésével a feladat nagyobb pontossággal végezhető, mint az önálló statisztikai alapú rendszerrel.

Először röviden bemutatjuk a megoldandó feladatot és a témában született legfontosabb eredményeket, majd a 3. fejezetben a parser építésének főbb lépéseit ismertetjük. A szabályalapú rendszer létrehozásánál kiindulási pontként Kornai környezetfüggetlen NP-nyelvtana (Kornai 1985, 1989) szolgált, melyet az NLTK nyelvtechnológiai programcsomag (Bird et al. 2009) segítségével implementáltunk. A Szeged Treebank (Csendes et al. 2005) alapján készült NP-korpusz tette lehetővé, hogy a parsert kiértékeljük, a munka egyes fázisaiban a legnagyobb hibaosztályokat elkülönítsük és a szabályrendszert ezek figyelembevételével fejlesszük – a 4. fejezet ezt a folyamatot írja le.

Végül az 5. fejezetben egy hibrid rendszert mutatunk be, mely a parser kimenetét használva gépi tanulási módszerrel végzi a magyar NP-k azonosítását, majd megmutatjuk, hogy két eszközünket egyesítve magasabb pontossággal tudjuk elvégezni a feladatot, mint a tisztán statisztikai alapú rendszerrel.

## 2. Előzmények

A szakirodalomban leggyakrabban *NP-chunking*-nak nevezett feladatnak (Abney 1991) alapján (Marcus 1994) adta azon definícióját, melyet felhasználva a feladat a nyelvtechnológiában használatos gépi tanuló algoritmusok egyik mércéjévé vált (ld. például a CoNLL 2000 versenyt, Tjong Kim Sang et al. 2000). Magas pontossággal végezték a feladatot többek között a Support Vector Machine (SVM), a Conditional Random Fields (CRF) vagy a Maximum Entropy Markov Model (MEMM) módszerrel is. Az utóbbi módszertől némiképp eltérően használ Maximum Entrópia tanulást és Rejtett Markov Modelleket saját NP-címkézőnk, a **hunchunk**, mely a feladatot magyar nyelvű szövegen is nagy pontossággal végzi ( $F_2 = 94.75\%$ ). A főnévi csoportok azonosításával kapcsolatban egy másik feladat, a maximális NP-k azonosítása

is megfogalmazható, mely egyes feladatokhoz, így különösen a frázis-alapú gépi fordításhoz fontos bemenetet szolgáltat. A **hunchunk** ezt a feladatot is 89 – 91% közötti F-pontszámmal végzi a különböző tesztadatokon.

Bár a feladat gépi tanulási módszerekkel nagy pontossággal oldható meg, feltételeztük, hogy egy kézzel írt szabályokon alapuló mondattani elemző (parser) a magyar főnévi csoportokat hasonló vagy magasabb pontossággal is képes azonosítani. Első kísérleteink megmutatták, hogy a mondatszintű nyelvtan hiánya nem teszi lehetővé, hogy az egyébként előnyben részesített „maximális NP” feladatot egy parser magas pontossággal megoldja, így a hagyományos, minimális NP-kre irányuló feladat minél nagyobb pontosságú teljesítését tűztük ki célul, bízva abban, hogy az így születő elemzések a statisztikai rendszer teljesítményén is javíthatnak.

### 3. A parser építése

#### 3.1. Formalizmus

A magyar NP-parser létrehozásához Kornai magyar NP-nyelvtanát használtuk. Az implementációhoz az NLTK programcsomagot választottuk, mivel különböző mondatelemző algoritmusok széles választékát támogatja és jegy-struktúrákra hivatkozó nyelvtani szabályok használatát is lehetővé teszi. A nyelvtan szabályai ugyanis nem csupán a szavak szófajára, hanem azok számos morfológiai jegyére is képesek hivatkozni. A **hunmorph** morfológiai elemző (Trón et al. 2005) lehetővé teszi, hogy az elemezni kívánt mondatok szavairól valamennyi ilyen információt kinyerjünk. A **hunmorph** a KR-kód (Rebrus et al. 2010) szerinti morfológiai elemzést ad, ezek pedig maguk is jegy-struktúrák, így mechanikusan alakíthatóak át a parser bemeneti formalizmusának megfelelő formátumra – ezáltal a morfológiai elemző valamennyi lehetséges kimenete megfelel a nyelvtan egy terminális szimbólumának. A nemterminális szimbólumok egy kategóriacímkekből (**NOUN**, **ADJ**, stb.) és a hozzátartozó jegy-érték struktúrából állnak.

A jegyek értéke lehet sztring, egész szám, újabb jegy-struktúra, más nemterminális szimbólum és a jegy értékét egy másik jegyértékhez kötő változó. Így például egy az ige és tárgy közötti számbeli egyezést kifejező szabály **VP** -> **V[PL=?a] N[PL=?a]** formátumú, mely egyenértékű a megszokottabb, görög betűket használó jelöléssel: **VP** -> **V[αPL] N[αPL]**. A nyelvtan a projekciós szinteket nem különböző szimbólumokkal, hanem a **BAR** jeggyel kódolja; így a **NOUN[BAR=0]** például egy pusztán főnevet jelöl. A morfológiai elemzés a képzett szavak esetében megadja a képzés forrását és típusát is; ezeket az **SRC** jegy kódolja, melynek értéke két további jegy, a tövet tároló **STEM** és a képzés típusát tartalmazó **DERIV**. Néhány példa az NLTK formalizmusa és a KR-kódolás közti megfeleltetésre (1)-ben látható.

- (1a) NOUN[POSS=[1=1, PLUR=1] -> NOUN<POSS<1><PLUR>>  
 (1b) NOUN[POSS=1, CAS=[SUE=1]] -> NOUN<POSS><CAS<SUE>>  
 (1c) NUM[CAS=[INS=1], SRC=[STEM=NUM, DERIV=ORD]] -> NUM[ORD]/NUM<CAS<INS>>  
 (1d) VERB[SUBJUNC-IMP=1, PERS=[1=1], PL=1, D=1] ->  
 -> VERB<SUBJUNC-IMP><PERS<1>><PLUR><DEF>  
 (1e) NOUN[POSS=1, SRC=[STEM=VERB[SRC=[STEM=VERB, DERIV=MEDIAL]], DERIV=GERUND]] ->  
 -> VERB[MEDIAL]/VERB[GERUND]/NOUN<POSS>

Noha a KR-kódok túlnyomórészt egyértelműen megfeleltethetőek ennek a reprezentációnak, néhány átalakítást mégis szükséges volt elvégezni. A KR-elemzés tartalmaz privatív jegyeket, például egy főnév egyes számát a <PLUR> jegy hiánya ‘jelöli’. Mivel szeretnénk, hogy a nyelvtan hivatkozhasson ezen jegyek hiányára, így az NLTK formátumában ezek a jegyek binárisak: a harmadik személyt, az egyes számot és a nominatív esetet rendre a PERS=0, CAS=0, PLUR=0 jegyek jelölik.

### 3.2. Főnévi csoportok azonosítása

Mivel a nyelvtan csupán a főnévi csoportokat felépítő szabályokat tartalmaz, így teljes mondatelemzést nem készíthetünk: az elemzés során bottom-up módszerrel azonosítjuk az olyan szószorozatokat a mondatban, melyet az NP-nyelvtan elfogad. A parser kimenete egy táblázat, melyből minden ilyen szóorra kiolvasható, hogy mely szabályok mely szósorokra való alkalmazásával épült fel az adott főnévi csoport. A mondatszintű nyelvtan hiánya azt jelenti, hogy nem zárhatunk ki lehetséges elemzéseket azért, mert azokból később nem építhető teljes mondatelemzés. Ehelyett néhány egyszerű szabállyal választjuk ki a végleges chunkolást, azaz a diszjunkt NP-szekvenciákat.

Kiindulópontként vesszük valamennyi, a parser által NP-ként felismert szószorozatot. Mivel minimális NP-ket keresünk, első lépésként kizárjuk azokat a sorozatokat, melyek egynél több főnevet tartalmaznak. Ezután, mivel a főnévi csoportok legmagasabb projekcióit keressük, kizárjuk azokat a jelölteket, melyeket egy másik intervallum tartalmaz. Következő lépésként az egymással átfedő jelöltek közül azokat választjuk, melyeket más kategóriájú frázisként a nyelvtan nem ismer fel – ezt az eljárást elsősorban az indokolja, hogy az eliptikus NP-ket engedélyező ún. SLASH-szabályok gyakran tévesen ismerik fel főnévi csoportként a mondat melléknévi és számnévi csoportjait. Ha ez az eljárás nem vezet sikerre – azaz mindkét vagy egyik jelölt sem áll elő más kategóriájú frázisként, akkor egyiket sem tartjuk meg.

## 4. A nyelvtan fejlesztése

A parser-t a Szeged Treebank alapján készült NP-korpusz segítségével értékeltük ki. Az eredeti NP-nyelvtan 81.76%-os F-pontszámot ért el az 1000 mondatból álló tesztkorpuszon. A parser teljesítményét a nyelvtanon végrehajtott minden változtatás után újra megmértük, a tévesen elemzett

mondatok szemrevételezésével pedig elkülönítettük a mindenkori legnagyobb hibaosztályokat.

Az első mérés során a legtöbb hibát nem a nyelvtan hiányossága okozta, hanem az a tény, hogy melléknévi és számnévi csoportokra hivatkozik, melyekre nem adtunk nyelvtant. Így első lépésként néhány egyszerű szabályt adtunk hozzá a nyelvtanhoz:

- (2a) ADJ -> ADJ ADJ
- (2b) ADJ -> ADV ADJ
- (2c) NUM -> NUM NUM
- (2d) NUM -> ADV NUM
- (2e) NUM -> ADJ NUM

Ezáltal a rendszer F-pontszáma 84.18%-ra nőtt.

A következő nagy hibaosztályt a névmások jelentették. Mivel a magyar névmások a főnevekhez nagyon hasonló módon viselkednek, így sem a nyelvtan, sem a *hunmorph* elemző nem tesz köztük különbséget. Így mind a KR-formalizmust, mind az NP-nyelvtant kibővítettük a PRON jeggel. Egyes esetekben szükségesnek bizonyult a névmások típusára is hivatkozni, így a jegy értékeként ezt is megadtuk. A két kategória egybeejtése ugyan általában megalapozott, mégis ez a lépés tette lehetővé, hogy a nyelvtan kezelje az általános és határozatlan névmásokat tartalmazó főnévi csoportokat (ld. 3)

- (3a) *minden pofon*
- (3b) *néhány villanykörte*

Ezeket a névmásokat a főnevek elsősztintű projekcióihoz csatoltuk, így a fenti esetek kezelésére a nyelvtant az alábbi szabályokkal bővítettük:

- (4a) NOUN[POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f] ->  
-> NOUN[PRON=GEN] NOUN[BAR=1, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f]
- (4b) NOUN[POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f] ->  
-> NOUN[PRON=INDEF] NOUN[BAR=1, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f]

Ez a módosítás az F-pontszámot 85.45-ra növelte.

Külön kezelést igényelt a mutató névmás egy speciális esete, melyre (5)-ben adunk példát:

- (5a) *ez a pincér*
- (5b) *ezek a hajók*
- (5c) *attól a pasastól*

Ezeket a szerkezeteket az (6)-beli szabállyal kezeljük, ezzel újabb egy százalékpontos javulást érve el az elemző teljesítményében.

- (6) NOUN[POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e] ->  
-> NOUN[PRON=DEM, BAR=0, POSS=?a, PLUR=?b, ANP=?c, CAS=?d]  
ART NOUN[PRON=0, BAR=2, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=0],

A következő jelentős hibacsoport olyan, főnevet módosító melléknévi csoportokhoz kötődött, melyeknek feje egy igéből képzett melléknév és tartalmazzák az eredeti ige argumentumát is (ld. 7).

(7a) *a korsónak támasztott könyvet olvasta*

(7b) *az ókori mór hódítóktól származó esküvést hallották*

Mivel a nyelvtan terminális szimbólumai kódolják a képzési információt, így lehetőség nyílt ezeket az eseteket a melléknévi csoportok nyelvtanába felvenni. A (7a) és (7b) alatti szerkezeteket rendre a (8a) és (8b) alatti szabályok kezelik.

(8a) ADJ -> NOUN ADJ [SRC=[STEM=VERB [], DERIV='PERF\_PART']]

(8b) ADJ -> NOUN ADJ [SRC=[STEM=VERB [], DERIV='IMPERF\_PART']]

Ez a módosítás a rendszer teljesítményét 87.87%-ra növelte. Mindezek után a parser hibáinak legnagyobb részét már a valóban kétértelmű szerkezetek okozták, azonban még számos hibát okozott az írásjelek és kötőszavak téves elemzése. Néhány, az ilyen elemek NP-beli és NP-környéki viselkedését leíró szabály a pontosságot további másfél százalékponttal növelte.

A nyelvtan végső változata, mely a cikk végén teljes egészében olvasható, a tesztadaton 89.36%-os F-pontszámot ér el. A teljesítmény javulását az egyes lépések függvényében az 1. táblázat foglalja össze. A fennmaradó hibák egy része mögött valódi szerkezeti kétértelműség áll, leggyakrabban azonban olyan hibaosztályokat találunk, melyek a korpusz sajátosságaiból fakadnak, így például gondot okoz egy-egy szövegtípusra jellemző speciális írásjelek szokatlan tokenizálása. Mivel a munka ezen fázisában egy-egy hasonló jelenség már a hibák nagyobb százalékáért felelős, mint bármelyik kezeletlen nyelvi szerkezet, úgy hisszük, hogy a pontosság további növelésének – mind a szabályalapú, mind a gépi taunoló eszköz esetében – feltétele a további tanuló- és tesztadat.

| Fejlesztés              | F-pontszám |
|-------------------------|------------|
| Kornai 1985             | 81.76%     |
| AdjP, NumP              | 84.18%     |
| Névmások                | 85.45%     |
| „Ez a” szerkezet        | 86.68 %    |
| Deverbális melléknevek  | 87.87%     |
| Írásjelek és kötőszavak | 89.36%     |

1. táblázat. A nyelvtan fejlesztésének lépései és hatásuk az elemzés pontosságára

A fenti eredmények alapján elmondhatjuk, hogy a nyelvtanba újonnan felvett szabályok az eredeti rendszer hibáinak közel felét kiküszöbölik. Az elért eredmény jelentősen elmarad a statisztikai alapú **hunchunk**-nak a minimális NP-k azonosításán elért eredményétől (94.75%), de lehetővé teszi, hogy megpróbálkozzunk hibrid rendszer kifejlesztésével.

## 5. Hibrid megoldás

A hibrid rendszer lényege, hogy a gépi tanulásra alapuló **hunchunk** rendszer tanításakor felhasználjuk a szabályalapú elemző kimenetét. Miután a parser a 3. fejezetben leírtak szerint elkészíti a mondat NP-chunkolását, azaz diszjunkt főnévi csoportokat jelöl meg, az egyes szavakat olyan címkékkel látja el, mint amilyenekre a statisztikai rendszert tanítottuk: a B-NP, I-NP, E-NP, 1-NP és 0 címkék jelölik rendre az NP elején, közepén és végén álló szavakat, az egyetlen szóból álló NP-eket valamint az NP-n kívüli tokeneket.

A **hunchunk** rendszer teljesítményét úgy próbáltuk javítani, hogy az általa használt szószintű jegyek közé felvettük a parser által adott chunk-címkéket. Ezáltal a statisztikai modellnek lehetősége nyílik olyan súlyt rendelni a parser által adott válaszokhoz, mely a legmagasabb pontosságú címkézéshez vezet.

A minimális NP-k azonosításához elvégeztük a teljes NP-korpusz elemzését, majd az elemzett adatot tanító- és tesztadatra bontottuk ugyanúgy, mint tettük azt a **hunchunk** kiértékelésekor. Az eredmények a 2. táblázatban láthatók.

|                               | Precision | Recall | F-score |
|-------------------------------|-----------|--------|---------|
| <b>hunchunk</b>               | 94.61%    | 94.88% | 94.75%  |
| <b>hunchunk+parser jegyek</b> | 95.29%    | 95.68% | 95.48%  |

2. táblázat. A parser-jegyek hatása a minimális NP-k azonosítására

Mint az a fenti táblázatból is látható, a parser kimenetének figyelembevétele a statisztikai rendszer hibáinak 15%-os csökkenéséhez vezetett. A parser által adott címkék ugyancsak hasznosnak bizonyultak a maximális NP-k azonosításakor (ld. a 3. táblázatot).

|                                 | Precision | Recall | F-score |
|---------------------------------|-----------|--------|---------|
| <b>hunchunk</b>                 | 89.34%    | 88.12% | 88.72%  |
| <b>hunchunk+parser features</b> | 89.46%    | 88.76% | 89.11%  |

3. táblázat. A parser-jegyek hatása a maximális NP-k azonosítására

## 6. Összefoglalás

Cikkünkben a magyar főnévi csoportok azonosításának egy szabályalapú megoldását mutattuk be. Kornai 1985-ös nyelvтанának továbbfejlesztésével a minimális NP-k azonosításán közel 90%-os F-pontszámot értünk el. Bár ez az eredmény – elvárásainkkal ellentétesen – nem közelítette meg a statisztikai rendszer pontosságát, alapjául szolgált egy hibrid rendszer megalkotásának. A Maximum Entrópiás Markov Modellt (MEMM) használó **hunchunk** rendszer,

miután a szabályalapú rendszer kimenetét is figyelembe veszi, a korábbinál lényegesen magasabb pontosságot ér el mind a minimális, mind pedig a maximális NP-k azonosításában.

## Hivatkozások

1. S. Abney. Parsing by chunks. *Principle-based parsing*, pages 257–278, 1991.
2. S. Bird, E. Klein, and E. Loper. *Natural language processing with Python*. O’Reilly Media, 2009.
3. D. Csendes, J. Csirik, T. Gyimóthy, and A. Kocsor. The Szeged Treebank. In *Lecture Notes in Computer Science: Text, Speech and Dialogue*, pages 123–131, 2005.
4. A. Kornai. The internal structure of Noun Phrases. *Approaches to Hungarian*, 1:79–92, 1985.
5. A. Kornai. A fonévi csoport egyeztetése [Agreement in noun phrases]. *Általános Nyelvészeti Tanulmányok*, pages 183–211, 1989.
6. M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of english: The Penn Treebank. *Computational Linguistics*, 19:313–330, 1994.
7. P. Rebrus, A. Kornai, and D. Varga. Egy általános célú morfológiai annotáció [a general-purpose annotation of morphology]. *Általános Nyelvészeti Tanulmányok*, 2010.
8. G. Recski, D. Varga, A. Zséder, and A. Kornai. Fonévi csoportok azonosítása magyar-angol párhuzamos korpuszban [Identifying noun phrases in a parallel corpus of English and Hungarian]. *VI. Magyar Számítógépes Nyelvészeti Konferencia [6th Hungarian Conference on Computational Linguistics]*, 2009.
9. E. F. Tjong Kim Sang, S. Buchholz, and K. Sang. Introduction to the CoNLL-2000 shared task: Chunking, 2000.
10. V. Trón, A. Kornai, G. Gyepesi, L. Németh, P. Halácsy, and D. Varga. Hunmorph: open source word analysis. In *Proceedings of the Workshop on Software*, pages 77–85. Association for Computational Linguistics, 2005.

## A. Az NP-parser nyelvtana

```

NOUN[POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f] ->
  NOUN[PRON=POS] NOUN[BAR=2, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f]
NOUN[POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e] ->
  NOUN[PRON=DEM, BAR=0, POSS=?a, PLUR=?b, ANP=?c, CAS=?d]
  ART NOUN[PRON=0, BAR=2, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=0]
NOUN[POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f] ->
  NOUN[PRON=GEN] NOUN[BAR=2, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f]
NOUN[POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f] ->
  NOUN[PRON=INDEF] NOUN[BAR=2, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f]
NOUN[BAR=1, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f] ->
  ADJ NOUN[BAR=0, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f]
NOUN[BAR=1, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f] ->
  NOUN[BAR=0, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f]

```

NOUN[BAR=1, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f]/NOUN[BAR=0] ->  
 ADJ NOUN[BAR=0, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f]/NOUN[BAR=0]  
 NOUN[BAR=1, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f]/NOUN[BAR=0] ->  
 NOUN[BAR=0, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f]/NOUN[BAR=0]  
 NOUN[BAR=2, POSS=?a, PLUR=0, ANP=?c, CAS=?d, D=?e, PRON=?f] ->  
 NUM NOUN[BAR=1, POSS=?a, PLUR=0, ANP=?c, CAS=?d, D=?e, PRON=?f]  
 NOUN[BAR=2, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f] ->  
 NOUN[BAR=1, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f]  
 NOUN[BAR=2, POSS=?a, PLUR=0, ANP=?c, CAS=?d, D=?e, PRON=?f]/NOUN[BAR=0] ->  
 NUM NOUN[BAR=1, POSS=?a, PLUR=0, ANP=?c, CAS=?d, D=?e, PRON=?f]/NOUN[BAR=0]  
 NOUN[BAR=2, POSS=?b, PLUR=0, ANP=?c, CAS=?d, D=?e, PRON=?f]/NOUN[BAR=0] ->  
 NOUN[BAR=1, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e]/NOUN[BAR=0, PRON=?f]  
 NOUN[BAR=3, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f] ->  
 ART[D=?e] NOUN[BAR=2, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, PRON=?f]  
 NOUN[BAR=3, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=1, PRON=?f] ->  
 NOUN[BAR=0, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=1, PRON=?f]  
 NOUN[BAR=3, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f]/NOUN[BAR=0] ->  
 ART[D=?e] NOUN[BAR=2, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, PRON=?f]/NOUN[BAR=0]  
 NOUN[BAR=3, POSS=0, PLUR=?a, ANP=?b, CAS=?c, D=1, PRON=?f] ->  
 NOUN[BAR=3, ANP=0, CAS=0] NOUN[BAR=2, POSS=1, PLUR=?a, ANP=?b, CAS=?c, PRON=?f]  
 NOUN[BAR=4, POSS=0, PLUR=?a, ANP=?b, CAS=?c, D=1, PRON=?f] ->  
 NOUN[BAR=3, CAS=[DAT=1]] NOUN[BAR=3, POSS=1, PLUR=?a, ANP=?b, CAS=?c, D=1, PRON=?f]  
 NOUN[BAR=3, POSS=0, PLUR=?a, ANP=?b, CAS=?c, D=1, PRON=?f] ->  
 ART[BAR=1, D=1, ME=?d, YOU=?e, PLUR=?f]  
 NOUN[BAR=2, POSS=[ME=?d, YOU=?e, PLUR=?f], PLUR=?a, ANP=?b, CAS=?c, PRON=?f]  
 NOUN[BAR=3, POSS=0, PLUR=?a, ANP=?b, CAS=?c, D=1, PRON=?f] ->  
 ART[BAR=0] NOUN[BAR=2, POSS=[], PLUR=?a, ANP=?b, CAS=?c, PRON=?f]  
 NOUN[POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f, BAR=?g] ->  
 PUNCT[TYPE='DQUOTE'] NOUN[BAR=?g, POSS=?a, PLUR=?b, ANP=?c, CAS=?d, D=?e, PRON=?f]  
 PUNCT[TYPE='DQUOTE']  
 NOUN/NOUN ->

ART[BAR=1, D=1, ME=?a, YOU=?b, PLUR=?c, PRON=?f] -> ART[D=1] PRO[ME=?a, YOU=?b, PLUR=?c, PRON=?f]  
 ART[D=1] -> DET  
 ADJ -> ADJ ADJ  
 ADJ -> ADV ADJ  
 ADJ -> NOUN ADJ[SRC=[STEM=VERB[], DERIV='PERF\_PART']]  
 ADJ -> NOUN ADJ[SRC=[STEM=VERB[], DERIV='IMPERF\_PART']]  
 ADJ -> PUNCT[TYPE='DQUOTE'] ADJ PUNCT[TYPE='DQUOTE']  
 ADJ -> ADJ PUNCT[TYPE=COMMA] ADJ  
 ADJ -> ADJ PUNCT[TYPE=COMMA] CONJ ADJ  
 NUM -> NUM NUM  
 NUM -> ADV NUM  
 NUM -> ADJ NUM