What does the Nom say? An algorithm for case disambiguation in Hungarian

Noémi Ligeti-Nagy
MTA-PPKE
Hungarian Language Technology Research Group
ligeti-nagy.noemi@itk.ppke.hu

Andrea Dömötör
MTA-PPKE
Hungarian Language Technology Research Group
domotor.andrea@itk.ppke.hu

Noémi Vadász
Research Institute for Linguistics
Hungarian Academy of Sciences
vadasz.noemi@nytud.mta.hu

Abstract

In this paper, we present our algorithm called nom-or-not designed for dissolving case-disambiguation in Hungarian. By case, we mean an abstract syntactic case, a kind of syntactic role of the given token. Nouns and proper names, adjectives, participles and numerals without a case suffix are always tagged as Nom, although the lack of case ending may represent various functions: it may mark the subject of the sentence or a possessor or the nominal part of a nominal predicate or the vocative case; on top of that, a modifier of a nominal or a nominal combined with a postposition lacks a case suffix as well; proper names consisting of two or more elements are also caseless. Our algorithm is motivated by the needs of a psycholinguistically motivated parser which aims to process sentences from left to right. Therefore, our case disambiguator follows the basic principles of the parser and analyses the sentences from left to right, always making a decision based on the information of the previously processed elements and the elements in a two token wide look-ahead parsing window. Our preliminary results show that if some modifications and new rules are added and it's run on a more precisely annotated corpus, it can improve the disambiguator algorithm. The preliminary results were obtained from a manually annotated corpora of 500 sentences.

Kivonat

Tanulmányunkban bemutatjuk a `nom-or-not` névre keresztelt algoritmust, amely az esetrag nélküli névszók mondatbeli szerepének egyértelműsítését végzi. A magyarban a testes esetrag hiánya nem egyértelműen a nominatívuszi esetet kódolja; egy testes esetragot magán nem viselő névszó többféle szerepet betölthet a mondatban: lehet valóban alany, lehet birtokos, lehet névszói állítmány névszói része, lehet névszó módosítója vagy névutó előtt álló névszói elem, lehet vokatívuszi esetben álló névszó, vagy lehet több elemből álló tulajdonnév egyik belső eleme. Algoritmusunk háttere egy az emberi szövegfeldolgozást modellálni szándékozó elemzőrendszer. Ennek szellemében az eset-egyértelműsítést az elemzőrendszer működési elvei alapján végezzük: balról jobbra haladva dolgozzuk fel a szöveget, mindig csak az eddig már olvasott szavakon, illetve a kételemű, előretekintő elemzési ablakban látható információra támaszkodva. Szabályalapú algoritmusunkat egy 500 mondatból álló korpuszon értékeljük ki, melyben kézzel annotálunk minden testes esetrag nélküli névszót. Eredményeink azt mutatják, hogy eljárásunk – két előzmény-algoritmusához hasonlóan, melyek egy-egy részfeladat kezelésére születtek – pontossága néhány szabály beillesztésével és pontosabban annotált korpusz használatával könnyedén javítható lehet.

1 Introduction

Here we present a rule-based algorithm offering a case disambiguation method designed primarily for the AnaGramma parsing system (Prószéky and Indig, 2015; Prószéky et al., 2016). Following a brief description of the parsing system and its principles we turn to the discussion of the possible functions of nominals in the sentence and finally we introduce our algorithm disambiguating caseless nominals in Hungarian sentences.

The current algorithm is an upgraded and extended version of two previously described algorithms presented in Ligeti-Nagy et al. (2018) and Dömötör (2018), to be described in details below.

It is inevitable to clarify some terminological questions. Throughout this article, by nominals we mean nouns, adjectives, participles and numerals – words able to fill the role of a subject or a possessor, words that may be modifiers of another nominal etc. This also applies to pronouns that substitute these lexical word classes.

With case, and more specifically, with *function in the sentence* on the one hand we refer to the set of syntactic roles in the sentence, on the other hand we use *function in the sentence* to identify nominal positions inside a noun phrase (NP) as well. Therefore, when we aim to specify the function of a caseless nominal in the sentence, we want to determine if it is the argument of a verb. Otherwise we specify its role inside an NP.

2 Background

Our algorithm fits into the frame of AnaGramma, a psycholinguistically motivated parsing system which tries to model human sentence processing with its left-to-right and word-by-word parsing design. The context of a word often influences its interpretation, therefore AnaGramma uses a two token wide look-ahead window that provides information of the right context of the word, while the information of the previously processed elements is always available in the so-called *pool.* The theoretical background of AnaGramma is based on a two-phased sentence processing model, the *Sausage Machine* where the parsing process consists of two main phases. The first phase is – as Frazier and Fodor (Frazier and Fodor (1978)) calls it – the *Preliminary Phrase Packager* which assigns the lexical and phrasal nodes to groups of words within the string input. The look-ahead window of AnaGramma implements this first phase. In this phase the components of the sentence are prepared, e.g. disambiguation of case-ambiguous nominals presented in this paper. Then, in the second phase, these packaged phrases get their roles in the sentence by adding non-terminal nodes. The second phase is called the *Sentence Structure Supervisor*, as the packages – the pieces of the sausage – receive their role in the sentence.

2.1 Caseless nominals in Hungarian

Our basic assumption about caseless nominals is that their function in a sentence may be one of the followings: subject (1a), unmarked possessor (1b), argument of a postposition[1] (1c), element in vocative case (where the noun or the pronoun is used to address a person directly, (1d)), modifier of another nominal (1e), part of a proper name consisting of more than one component (1f) or part of a nominal predicate (discussed in detail later). In the examples (1a)-(1f), the third row illustrates the current morphological annotation of the words.

(1) a.

A	***szomszéd***	*tegnap*	*érkez-ett.*
the	neighbour	yesterday	arrive-Pst.3sg
Det|art.Def	**N.Nom**	Adv	V.Pst.NDef.3sg

‘The neighbour arrived yesterday.’

b.

A	***szomszéd***	*kutyá-ja*	*ugat.*
the	neighbour	dog-Poss.3sg	bark.3sg
Det|art.Def	**N.Nom**	N.Poss.3Sg.Nom	V.Prs.NDef.3sg

‘The neighbour’s dog barks.’

c.

A	***szomszéd***	*után*	*érkez-t-ünk.*
the	neighbour	after	arrive-Pst-1pl
Det|art.Def	**N.Nom**	Post	V.Prs.NDef.1sg

‘We arrived after the neighbour.’

d.

Jó	*reggel-t,*	***szomszéd!***
good	morning-Acc	neighbour
Adj.Nom	N.Acc	**N.Nom**

‘Good morning, neighbour!’

e.

Az	***előző***	*szomszéd*	*kedves*	*volt.*
the	previous	neighbour	kind	be.Pst.3sg
Det|art.Def	**Adj.Nom**	N.Nom	Adj.Nom	V.Pst.NDef.3sg

‘The previous neighbour was kind.’

[1] Considering the noun to be the argument of the postposition is a simplification and is motivated by AnaGramma being a dependency-parser. In our system, the noun is a dependent of the postposition. A detailed argument for this, however, is beyond the scope of this paper.

f. ***Máris*** *szomszéd* *tegnap* *érkez-ett.*
Máris neighbour yesterday arrive-Pst.3sg
N.Nom N.Nom Adv V.Pst.NDef.3sg

‘Neighbour Máris arrived yesterday.’

As mentioned above, the emphasised nominals in (1a)-(1f) are either arguments of the verb or parts of an NP. The role of predicative nominals, however, is different. Sentence parsing in computational linguistics is usually based on the verb and its argument frame. However, in Hungarian, it is possible to make well-formed, complete and non-elliptic sentences without a finite verb. This is due to the so-called zero copula phenomenon. The copula in Hungarian can be defined as ‘an expletive‘ which is present “if and only if its presence is required by a morphophonological constraint” (É. Kiss (2002):72). This morphophonological constraint is related to the Third Person Parameter of Stassen (1994) which means that the (verbal) copula can only be omitted in the third person in present tense. See examples in (2).

(2) a. *A* *szomszéd-om* *ügyvéd.*
the neighbour-Poss.1sg lawyer

‘My neighbour is a lawyer.’

b. *A* *szomszéd-om* *ügyvéd* *volt.*
the neighbour-Poss.1sg lawyer be.Pst.3sg

‘My neighbour was a lawyer.’

c. *Ügyvéd* *vagyok.*
lawyer be.1sg

‘I am a lawyer.’

In nominal sentences like (2a) the parsing tool needs to be able to identify whether or not the sentence contains a nominal predicate as the nominal this should be the head of the whole sentence.

Predicative nominals can be nouns, adjectives, occasionally numerals, and pronouns that substitute these elements. If the predicate is a noun phrase, then the nominal, which is the head of the phrase, is considered predicative. According to Higgins (1973) and others there are various types of copular clauses. The two main categories are predicative and equative sentences. By the former one, we mean those copular sentences in which the nominal predicate is a bare noun or adjective, an indefinite noun phrase or adjectival phrase which denotes an attribute of the subject. In contrast, the latter sentence type states the equality of two individuals. In this case, both the predicate and the subject must be a referential, therefore a determiner phrase (DP) – which encodes definiteness in Hungarian. DPs can be NPs with definite article, proper names or possessive NPs.

The predicative nominal is morphologically unmarked in both types of copular sentences, therefore it shows no difference in form to nominative, genitive and caseless nominals mentioned above. The issue of the case of predicative nominals could be subject of theoretical debate. There are three main approaches to this question. We could either 1) assume a phonologically zero predicative nominal, 2) claim that the nominal predicate is nominative which gets its case by the agreement with its subject (Szécsényi, 2000), or 3) simply consider it caseless. This study does not intend to take sides in this issue. With respect to automatic parsing, we may consider the predicative role as a functional feature indicating that the nominal in question needs a subject and optionally a copula as complement. The question of abstract cases does not have special importance for our purposes, therefore in the following, predicative nominals will simply be considered nominals with predicative feature and without case marking.

Based on the above described roles it can be stated that caseless nominals either bear a phonologically zero case suffix (when functioning as a subject as in (1a), an unmarked possessor (1b) or being in vocative case (1d)) or bearing nothing (when modifying another nominal (1e), being followed by a postposition (1c), being part of a complex proper name (1f), or functioning as a predicative nominal (2)).

Throughout this paper the following annotation is used to distinguish the previously detailed functions:

- phonologically zero case suffix (marking either a `Nom` or a `Gen`): α
- real 'caselessness' (marking the true lack of case ending): 0
- tag marking the predicative nominal: `Pred`
- `Nom`, `Gen`, and `Voc` stands for the nominative, genitive, and vocative case, respectively

2.2 Previous algorithms

`nom-or-what` presented in (Ligeti-Nagy et al., 2018) is a rule-based algorithm primarily built on the ideas drawn up in (Vadász and Indig, 2018) where a basic method was introduced to identify unmarked possessors in a sentence (`nomorgen`). `nom-or-what` aims to identify the roles of caseless nominals in the sentence solely based on the information seen in the two token wide look-ahead window and was essentially planned to be a module of the AnaGramma parser. This algorithm does not disambiguate nouns in vocative case and only operates on sentences with a verb present (meaning that it does not intend to identify nominal predicates). The algorithm was designed by analysing the caseless nominals both inside and in the final position of an NP in a syntactically annotated corpus of Hungarian (Szeged Treebank 2.0, Csendes et al. (2005)).

Three sets of rules were created for nouns, adjectives, and numerals. The rules attempt to define the precise role of the token under examination based on its morphological annotation and on the information gained from the parsing window. If no solid decision can be made and more information is needed to further specify an element, default tags were used. The performance of the algorithm was evaluated on a manually annotated corpus of 500 sentences collected from the Hungarian Gigaword Corpus (Oravecz et al., 2014). The high precision (97.73%) indicated that the basic principles of the algorithm are correct. The relatively low recall (67.63%) was explained by the authors as a sign of the excessive use of default values.

The algorithm described in Dömötör (2018) (named `is-pred`) was also designed to constitute a part of the AnaGramma parser. It follows the principles of the sausage machine model described in section 2. `nom-or-what` was basically the implementation of the first phase of this two-phased parsing model. The second phase carried out by `is-pred` uses the whole left context, the so-called *pool*. Besides, the `is-pred` algorithm strongly relies on the output of `nom-or-what`, as there would be little chance to identify predicative nominals based exclusively on the left context without taking into account the local decisions of the first phase.

In sum, the input of `is-pred` is a sequence that consists of the nominal in question and the part of the sentence that precedes it. The left context of the current word is already analysed and disambiguated by `nom-or-what` (if possible), thus the algorithm can use various pieces of morphosyntactic information from the pool. The output is a value, similar to trivalent logic: `Pred` if the nominal is obviously a predicate, `Nonpred` if it is obviously not a predicate, and `Undefined` if its syntactic role is still unclear from the given information.

The `is-pred` algorithm achieved high precision on its test, however, it has some deficiencies that should be improved. On the one hand, its responses are binary which do not complete the analysis in the `Nonpred` cases. On the other hand, `is-pred` only handles the predicative copular clauses, therefore the recognition of nominal predicates in equative sentences is a significant gap that this study intends to fill.

The idea behind the current algorithm – called `nom-or-not`, referring to its role as a synthesis of its antecedents – is, on the one hand, to merge all working and tested rules of the previous algorithms, and, on the other hand, to fill as many gaps left as possible.

3 Method

The method of `nom-or-not` follows `nom-or-what` and `is-pred` in being rule-based which means that the algorithm does not use machine learning approaches rather it is built on linguistically grounded, hand-crafted rules. The main difference among the three is that `nom-or-not` merges the two phases of parsing and aims to disambiguate each possible role of caseless nominals in one step. For this task, it is necessary to use both the window and the pool at the same time, therefore the algorithm operates with both forward- and back-looking rules. In either case, the principal source of information is the morphological annotation with only a small scent of lexical information. That is, the disambiguation of caseless nominals is carried out primarily based on the syntactic structure.

The algorithm is designed to process sentences annotated by the *emMorph* morphological analyser (Novák (2003), Novák (2014), Novák et al. (2016)), where the token, the lemma and the morphological tags are separated by a /, and the morphological tags are in square brackets (*USA/USA/[/N][NOM]*). The algorithm processes the sentences from left to right, word by word. The rules are only applied if the token under examination is tagged as `Nom`. As the targeted parsing method has a psycholinguistic motivation, the case disambiguation algorithm first gathers all the information of the given nominal that is deductible from the pool (the collection of the information of the already processed elements). The back-looking rules are used for preliminary disambiguation of predicative nominals, and they are the following:

- If there is a non-copular finite verb in the pool → the current token is not `Pred`
- If there is a nominative in the pool → the current token is `Pred`, if other cases will be ruled out based on the window, and only `Nom` and `Pred` remains as an option
- If the word is the possible head of a DP and there is no nominative in the pool → it is not `Pred`

  - If proper name → Head of DP
  - If possessive → Head of DP
  - If preceded by a determiner and optionally one or more NP-modifiers → Head of DP
  - If demonstrative pronoun (‘this’, ‘that’) → Head of DP

The rule to detect vocative case on nominals is rather simple at the moment: if the pool contains a verb in 1st or 2nd person singular or plural, and now we see a 3rd person singular or plural noun assigned nominative case Nom (see example (3), where the morphological annotation of the words can be seen in the third line), then it is a Voc.

(3)	*Jövök,*	***apám!***
	come.PRS1SG	father.POSS3SG
	[/V][Prs.NDef.1Sg]	[/N][Nom]

I’m coming, father!

Having exploited the left context the algorithm refines its judgment about the nominal in question using the information gathered from the window. The forward-looking rules are displayed in decision trees in Figures 1-3. Obviously, only those branches will be activated that are relevant considering the conclusions drawn up from the information coming from the pool; and every non-final decision is finalised if the knowledge based on the pool makes it possible to rule out a part of the outcome. (E.g. an edge leads us to a leaf with the tag *nom_or_pred* on it, but the pool already made it clear that the actual token cannot be a Pred, therefore here the tag Nom will be assigned to this token.)

As the algorithm does not exploit the whole sentence, there necessarily may remain cases where no certain decision can be made. We use the following tags for these cases:

- α: this is the default case of nouns; if no final tag can be assigned to a noun, but the predicative function is ruled out, the token in question is marked with an α, which can later be further specified as Nom or Gen
- *Nom/Pred*: a tag signaling that the given word may either be the subject of the sentence or the nominal predicate
- 0/*Pred*: a tag signaling that the given word may either be a modifier element in an NP or the nominal predicate of the sentence

We assume that nouns and proper names share the same default role – the one marked by α – with the default case of adjectives, participles, and numerals being 0.

Figure 1 shows the forward-looking rules activated when the token in question is a noun, a proper name, or a plural adjective or participle. The root of the tree is the POS-tag of the given word. The edges on the first level of the tree contain information gathered from the first element in the parsing window. As an example, if the first token in the window contains the tag of a postposition (Post), the algorithm assigns the tag 0 to the word in question, deleting its original Nom tag. The edges on the second level contain information seen on the second element in the parsing window. These edges are only activated if no final decision could be made based on the first token in the window and only a default tag was assigned to the given token.

A special distinction is made during the process not visible on the decision trees. If the given word has a possessive case suffix on it, no `Gen` tag can be assigned to it based on a possessive suffix on the second element in the window. It is a simplification with which we intend to rule out cases like (4a). The genitive case of a word like the one in bold in (4b) remain identifiable for the algorithm by detecting the possessive suffix on the first element in the window (*megbízásából* ('on behalf of the government')).

(4)	a.	*Magyarország*	***kormány-a***	*mostani*	*nyilatkozat-á-ból*
		Hungary	government-Poss.3sg	current	statement-Poss.3sg-out.of
		N.Nom	**N.Poss.3sg.Nom**	Adj.Nom	N.Poss.3sg.Ela

'from the current statement of the government of Hungary'

b.	*Magyarország*	***kormány-a***	*nyilatkozat-á-ból*
	Hungary	government-Poss.3sg	statement-Poss.3sg-out.of
	N.Nom	**N.Poss.3sg.Nom**	N.Poss.3sg.Ela

'from the statement of the government of Hungary'

The rules for singular adjectives and participles are displayed in Figure 2. The same distinction explained above is valid for the analysis of these tokens as well. Finally, the rules for numerals can be seen in Figure 3. Throughout the figures we use the macro `NPMod` for adjectives and participles as `nom-or-gen` started to tag every adjective and participle as `NPMod` referring to their ability to modify an NP.

The algorithm implemented in Python is available with the test corpus containing the gold standard annotation at `https://github.com/ppke-nlpg/nom-or-not`.

4 Results

For the evaluation of the performance of the algorithm we used a randomly composed subcorpus of the Hungarian Gigaword Corpus. The test corpus contains 500 sentences with no restriction to genre, content or quality. We carried out the morphological analysis of the sentences with the *emMorph* tool integrated in `e-magyar` language processing system (Váradi et al. (2018)).

The testcorpus contains 2 255 tokens tagged as `Nom` by the morphological analyser. We manually annotated them with tags from the set described above. The output of the algorithm was compared to this gold standard. It is important to note that the human annotation took the whole sentence into consideration and no default tags were allowed (unless the whole sentence was ambiguous). As the algorithm operates without analysing the whole sentence, it necessarily provides ambiguous responses in some cases meaning that we cannot expect 100% recall. The algorithm was consciously designed to work with high precision instead of high recall.

The evaluation follows the rules described in Table 1. The true positive (TP) matches are the correct ones. The erroneous or overspecified results are considered false positives (FP). Finally, we refer to the uncertain (underspecified) responses of the algorithm as false negatives (FN). The results are shown in Table 2. By precision we mean the percentage of $TP/(TP + FP)$ and by recall we mean $TP/(TP + FP + FN)$.

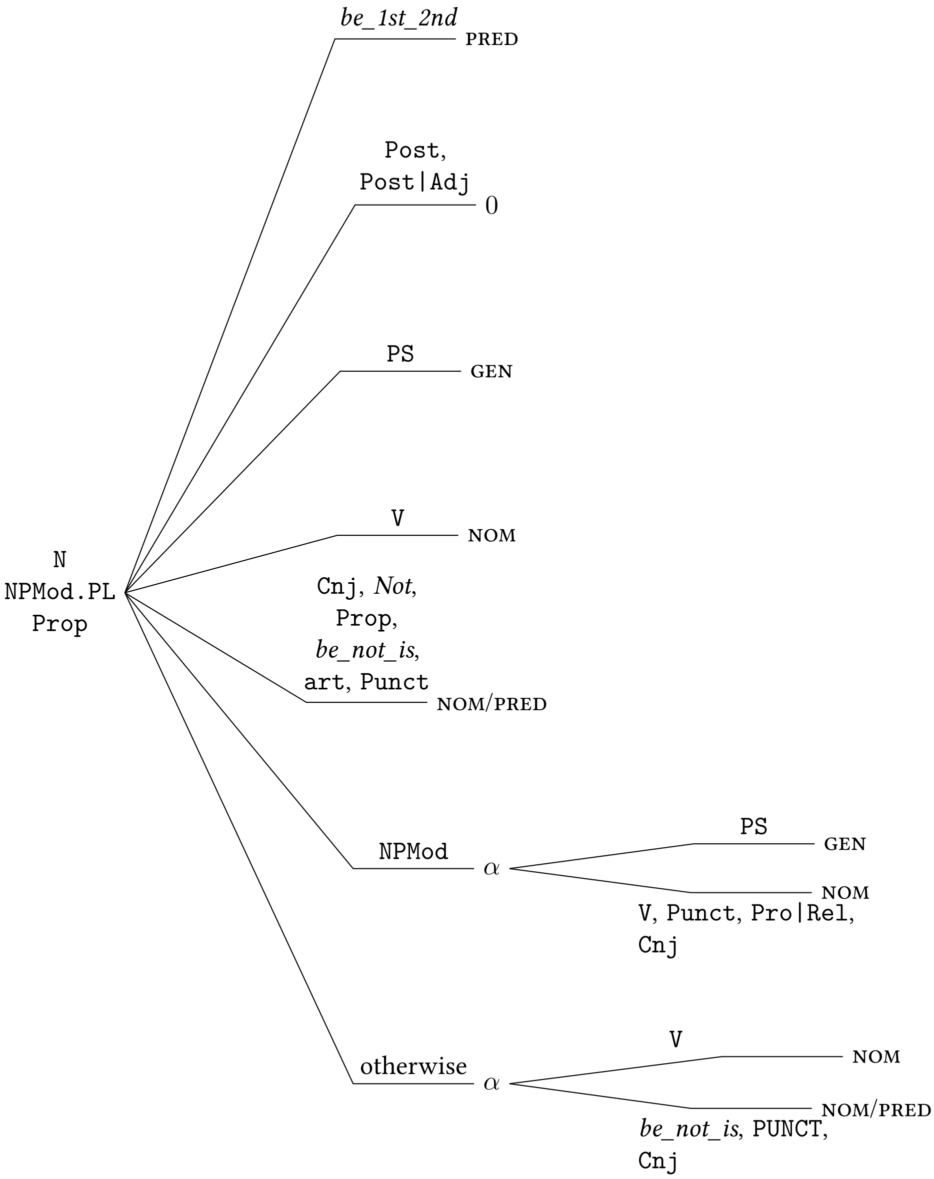

Figure 1: Decision tree summarising the rules concerning nouns, proper names, and plural adjectives, numerals and participles. The root of the tree is the POS-tag of the token under examination. The edges on the first level contain information seen on the first element in the parsing window. The edges on the second level contain information seen on the second element in the parsing window. *be_1st_2nd* is a macro for the 1st and 2nd person forms of the copula. *Not* is a macro for negation. *be_not_is* is a macro for any copula except for the singular and plural 3rd person form of *be*.

As can be seen, the algorithm achieved moderately good recall and a precision lower than expected. We analysed the results in more detail in a confusion matrix (Table 3). The rows display the responses of the algorithm, while the columns show

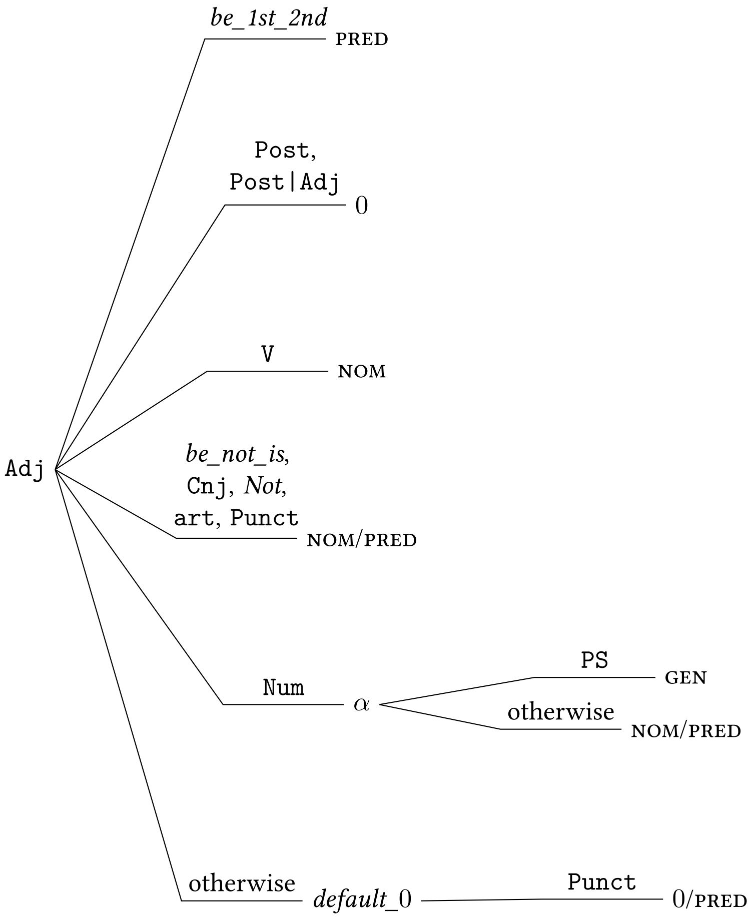

Figure 2: Decision tree summarising the rules concerning (singular) adjectives and participles. The root of the tree is the POS-tag of the token under examination. The edges on the first level contain information seen on the first element in the parsing window. The edges on the second level contain information seen on the second element in the parsing window. *be_1st_2nd* is a macro for the 1st and 2nd person forms of the copula. *Not* is a macro for negation. *be_not_is* is a macro for any copula except for the singular and plural 3rd person form of *be*.

the gold standard annotation.

A significant part of the errors (102) is due to an erroneous morphological annotation of the surrounding tokens. We eliminated those from the final results.

5 Discussion

As expected, the algorithm performs with a moderately high recall compared to that of `nom-or-what` (67.63%) which is due to the fact that some of the default tags were eliminated from the algorithm. Recall is influenced by the number of false negative hits (361 in the results). Considering that the algorithm does not have the whole sentence available when deciding, underspecification (resulting in false negative hits)

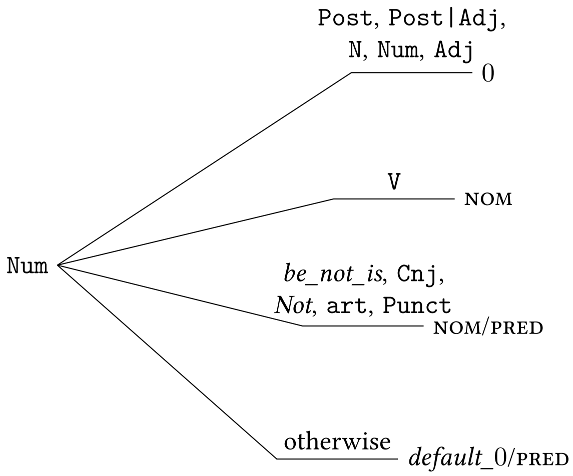

Figure 3: Decision tree summarising the rules concerning numerals. The root of the tree is the POS-tag of the token under examination. The edges on the first level contain information seen on the first element in the parsing window. *be_not_is* is a macro for any copula except for the singular and plural 3rd person form of *be*.

category	result	standard
TP	NOM	NOM
	GEN	GEN
	PRED	PRED
	0	0
	VOC	VOC
	α	α
FP	NOM	α
	GEN	α
	0	0/PRED
	PRED	0/PRED
	NOM	NOM/PRED
	PRED	NOM/PRED
	every other non-matching tags	
FN	α	NOM
	α	GEN
	0/PRED	0
	0/PRED	PRED
	Nom/Pred	Nom
	NOM/PRED	PRED

Table 1: Rules of evaluation. The tags in the *result* column are the ones assigned by the algorithm. The tags in the *standard* column are the gold standard annotation.

is comprehensible in many of the cases. These results are not as problematic for the whole parsing task as the false positive ones, since the uncertain tags can still be specified at a later point of parsing with the scanning of further words.

Precision	Recall	F-measure
77.82%	79.3%	78.55%

Table 2: Test results of the `nom-or-not` algorithm evaluated on 500 randomly selected and manually annotated sentences

	`Nom`	`Gen`	0	`Pred`	`Voc`	α	*Nom/Pred*
`Nom`	281	9	54	57	0	0	0
`Gen`	3	229	8	0	0	0	0
0	1	1	811	3	0	0	0
`Pred`	105	0	2	62	0	0	5
`Voc`	0	0	0	0	0	0	0
$alpha$	83	27	79	5	0	0	0
Nom/Pred	143	2	26	69	1	1	0
0*/Pred*	0	0	39	0	0	0	0

Table 3: Confusion matrix. The rows refer to the tags assigned by the algorithm. The columns represent the gold standard annotation.

The confusion matrix in Table 3 reveals that the majority of FP hits (268) is in connection with `Nom` or `Pred`, and more than half of them (162) is caused by a swap of these two tags. This can be explained on the one hand with the fact that our rules detecting predicative nominals are highly dependent on our preceding decisions on nominals: if a `Nom` was found we assume that no more `Nom` should be identified. However, our rules do not take clause boundaries into consideration, although a previously found `Nom` may be the subject of another clause other than the clause under examination . Stopping the backwards-looking rules on clause boundaries is a rather important issue to solve later. Obviously, any erroneously annotated `Nom` can lead to further mistakes during the analysis, even within the same clause. On the other hand, transposing `Nom` with `Gen` or vice versa is often caused by a verb falsely considered a copular verb. *Lehet* (may be) or *lesz* (will be) are just two examples of verbs that can either be a copular verb or a normal verb. This distinction is not available in their current morphological annotation, therefore the algorithm always assumes them to be a copular verb.

Another source of errors (159 cases) is the undiscovered inner structure of constructions like (5a) and (5b). Here we assume that there is no case suffix on the first element, therefore a 0 would be the correct tag for it. However, detecting these names is challenging and currently not solved in `nom-or-not`. There is no visible sign of the connection between the words in these constructions, especially not in their morphological analysis. Therefore, the first element most often receives a default tag. Presumably, these cases should be referred to a module responsible for world knowledge.

(5) a. *elnök* *úr*
president sir
N.Nom N.Nom

'Mr. President'

b. *Kinaesthetics termék*
Kinaesthetics product
Prop.Nom N.Nom

'the product Kinaesthetics'

Finally, cases like (6a) present a challenge to our algorithm as well: these are some kind of exclamations without any particular case suffix on them, as they play no role in the sentence. We would assign a 0 tag for them, but their distinction is quite problematic and at the moment unsolved in a sentence.

(6) a. *Támadás!*
attack
N.Nom

'Attack!'

Setting the unsolved problems and all the errors aside, we can see that the algorithm performs well with genitive case and with tokens not bearing any suffix at all (tagged with 0). With `Pred`, on the other hand, `nom-or-not` is quite uncertain, but never assigns any `Gen` or 0 tag to the nominal predicates of a sentence.

A part of the underspecification (FN results) may be solved by inserting a final step at the end of the analysis of each sentence: any verb following the tokens tagged as *Nom/Pred* can clarify its role as `Nom`.

6 Conclusion

We presented our rule-based algorithm called `nom-or-not` designed to disambiguate the role of caseless nominals for Hungarian. It is the successor of some related algorithms, each of which were implemented to solve a small part of the complex problem. Here we intended to provide an algorithm able to deal with every possible role of caseless nominals.

In this paper, we presented the design of the algorithm accompanied by the preliminary results obtained by evaluating the algorithm's performance on a test corpus containing 500 manually annotated sentences. Although we expected a higher precision, the majority of FP results is not a random mistake but a systematic error that can and should be solved by extending our rules or by evaluating the algorithm on a more precisely annotated test corpus. The recall is higher than our expectations proving that eliminating the default tags of adjectives, participles and numerals results in a better performance.

There are numerous tasks ahead of us: we need to revise our rules concerning predicative nominals as they seem to cause a significant amount of FP results. After inserting a final check in the algorithm that makes it able to clarify the role of tokens temporarily annotated with a tag of a default value, `nom-or-not` will hopefully provide a solution of high precision and recall for this case-disambiguation task for Hungarian.

References

Dóra Csendes, János Csirik, Tibor Gyimóthy, and András Kocsor. 2005. The Szeged Treebank. In Václav Matoušek, Pavel Mautner, and Tomáš Pavelka, editors, *Text, Speech and Dialogue: 8th International Conference, TSD 2005, Karlovy Vary, Czech Republic, September 12-15, 2005. Proceedings*. Springer Berlin Heidelberg, Berlin, Heidelberg, pages 123–131.

Andrea Dömötör. 2018. Nem mind VP, ami állít – A névszói állítmány azonosítása számítógépes elemzőben [All that Predicates is not VP – The Identification of Nominal Predicate in Automatic Parsing]. In Zsófia Ludányi, Valéria Krepsz, and Tekla Etelka Gráczi, editors, *Doktoranduszok tanulmányai az alkalmazott nyelvészet köréből 2018*. pages 3–10.

Katalin É. Kiss. 2002. *The Syntax of Hungarian*. Cambridge University Press.

Lyn Frazier and Janet Dean Fodor. 1978. The Sausage Machine: A New Two-stage Parsing Model. *Cognition* 6(4):291–325.

Francis Roger Higgins. 1973. *The Pseudo-Cleft Construction in English*. Garland Press, New York.

Noémi Ligeti-Nagy, Noémi Vadász, Andrea Dömötör, and Balázs Indig. 2018. Nulla vagy semmi? Esetegyértelműsítés az ablakban [Zero or Nothing? Case Disambiguation in the Window]. In Veronika Vincze, editor, *XIV. Magyar Számítógépes Nyelvészeti Konferencia*. pages 25–37.

Attila Novák. 2003. Milyen a jó Humor? [What is Good Humor Like?]. In *I. Magyar Számítógépes Nyelvészeti Konferencia*. SZTE, Szeged, pages 138–144.

Attila Novák. 2014. A New Form of Humor – Mapping Constraint-Based Computational Morphologies to a Finite-State Representation. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland.

Attila Novák, Borbála Siklósi, and Csaba Oravecz. 2016. A New Integrated Open-source Morphological Analyzer for Hungarian. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), Paris, France.

Csaba Oravecz, Tamás Váradi, and Bálint Sass. 2014. The Hungarian Gigaword Corpus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland.

Gábor Prószéky and Balázs Indig. 2015. Magyar szövegek pszicholingvisztikai indíttatású elemzése számítógéppel [Psycholingvistically Motivated Analysis of Hungarian Texts with Computer]. *Alkalmazott Nyelvtudomány* 15(1-2):29–44. Original document in Hungarian.

Gábor Prószéky, Balázs Indig, and Noémi Vadász. 2016. Performanciaalapú elemző magyar szövegek számítógépes megértéséhez [A Performance-based Parser to the Comprehensive Understanding of Hungarian Texts]. In Kas Bence, editor, *"Szavad ne feledd!": Tanulmányok Bánréti Zoltán tiszteletére*, MTA Nyelvtudományi Intézet, Budapest, pages 223–232. Original document in Hungarian.

Leon Stassen. 1994. Typology Versus Mythology: The Case of the Zero-Copula. *Nordic Journal of Linguistics* 17(2):105–126.

Tibor Szécsényi. 2000. Esetegyeztetés a predikatív főnévi csoportban [Case agreement in the predicative noun phrase]. In László Büky and Márta Maleczki, editors, *A mai magyar nyelv leírásának újabb módszerei IV.*, Szegedi Tudományegyetem, Szeged, pages 189–202.

Noémi Vadász and Balázs Indig. 2018. A birtokos esete az ablakkal [Possessor's Case with the Window]. In György Scheibl, editor, *LingDok: Nyelvész-doktoranduszok dolgozatai 17.*, SZTE Nyelvtudományi Doktori Iskola, Szeged, pages 85–99.

Tamás Váradi, Eszter Simon, Bálint Sass, Iván Mittelholcz, Attila Novák, Balázs Indig, Richárd Farkas, and Veronika Vincze. 2018. E-magyar – A Digital Language Processing System. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA), Miyazaki, Japan.