magyarlanc: A Toolkit for Morphological and Dependency Parsing of Hungarian

János Zsibrita[1], **Veronika Vincze**[1,2] and **Richárd Farkas**[1]
[1]Department of Informatics, University of Szeged
{zsibrita,rfarkas}@inf.u-szeged.hu
[2]Hungarian Academy of Sciences
Research Group on Artificial Intelligence
vinczev@inf.u-szeged.hu

Abstract

Hungarian is the stereotype of morphologically rich and free word order languages. Here, we introduce magyarlanc, a natural language toolkit developed for the linguistic preprocessing – segmentation, morphological analysis, POS-tagging and dependency parsing – of Hungarian texts. We hope that the free availability of the toolkit fosters the research not just on the Hungarian language but on all the morphologically rich languages in general. The main novelties of the tool are the application of a new harmonized morphological coding system of Hungarian, the data-driven approach and the integration of a dependency parser. The system is implemented in JAVA, hence it can be used in a platform-independent way.

1 Introduction

For end user natural language processing applications, it is essential to have access to a basic linguistic analyzer tool on the target language, in order to prevent reinventing the wheel every time. In this paper, we present magyarlanc, a basic linguistic analyzer toolkit developed for Hungarian.

Hungarian is a morphologically rich language with free word order (i.e. leaving aside the issue of the internal structure of NPs, most sentence-level syntactic information in Hungarian is conveyed by morphology, not by configuration). A large part of the methodology for morphosyntactic analysis has been developed for English. However, the linguistic analysis of morphologically rich and free word order languages requires special techniques. Hence, it was not sufficient to simply employ available tools and retrain on Hungarian corpora, we had to modify/adapt them. We hope that our findings and experiences gained during this adaptation process are useful for everybody dealing with morphologically rich – especially agglutinative – languages.

magyarlanc is enriched with a sentence splitter and tokenizer, a morphological analyzer, a POS-tagger and a dependency parser, each of them fine-tuned for the characteristics of Hungarian. The main novelties of magyarlanc are the following (each of the three criteria is unique among Hungarian-oriented linguistic analyzers):

- It is data-driven. Every module was systematically trained and evaluated on the Szeged Corpus and Szeged Dependency Treebank (82K sentences with manual annotation).
- It is an integrated toolkit, starting from raw text outputs to dependency parses.
- It is implemented fully in JAVA (incorporation to big systems is straightforward).

magyarlanc is freely available for research purposes at http://www.inf.u-szeged.hu/rgai/magyarlanc.

The structure of the paper is the following. First, we provide a summary of the grammatical features of Hungarian, which is followed and a short description of Hungarian morphological coding systems. Then we present the modules of magyarlanc. We also test the efficiency of magyarlanc and provide results on morphological and dependency parsing.

2 Grammatical Features of Hungarian

In this section, we provide a basic description of the Hungarian language with special emphasis on the phenomena that are important for morphological and syntactic parsing, based on Farkas et al. (2012). For a better understanding of the phenomena described, English will be used as a contrast language.

Proceedings of Recent Advances in Natural Language Processing, pages 763–771,
Hissar, Bulgaria, 7-13 September 2013.

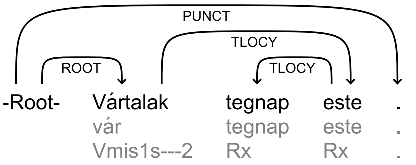

Figure 1: Dependency graph of the sentence *Vártalak tegnap este* “I was waiting for you last night”.

Hungarian is an agglutinative language, thus a word can have hundreds of word forms due to inflectional or derivational affixation. Grammatical information is usually encoded in morphology and Hungarian is a typical morphologically rich language. Word order is free in the sense that the positions of the subject, the object and the verb are not fixed within the sentence, but word order is related to information structure, e.g. new (or emphatic) information (the focus) always precedes the verb and old information (the topic) precedes the focus position. Thus, the position relative to the verb has no predictive force as regards the syntactic function of the given argument: while in English, the noun phrase before the verb is most typically the subject, in Hungarian, it is the focus of the sentence, which itself can be the subject, object or any other argument (É. Kiss, 2002).

The grammatical function of words is determined by case suffixes as in *lánc* “chain” – *lánccal* (chain-INS) “with (a/the) chain”. Hungarian nouns can have about 20 cases[1] and – being a head-final language – case suffixes always occur at the right end of the word as in *lánc* “chain” – *láncaikkal* (chain-3PLPOSS-PL-INS) “with their chains”. Case suffixes mark the relationship between the head and its arguments (subject, object, dative etc.).

Verbs are inflected for person and number and the definiteness of the object. Conjugational information is sufficient to deduce the pronominal subject or object, hence they are mostly omitted from the sentence: *Vártalak tegnap este.* (wait-PAST-1SG2OBJ yesterday evening) “I was waiting for you last night”. This pro-drop feature of Hungarian leads to the fact that there are several clauses without an overt subject or object, however, the first person singular subject and the second person object can be reconstructed on the basis of the grammatical features of the verb (see Figure 1).

Hungarian is characterized by vowel harmony, which means that most of the suffixes exist in two different forms – one with a front vowel and another one with a back vowel – and it is the vowels within the stem that determine which form of the suffix is attached to the word. For instance, the verb *fut* “run” is inflected as *futnak* “they run” in the third person plural because the stem contains a back vowel but the same form of the verb *mer* “dare” is *mernek* “they dare” since there is a front vowel in the stem.

There are several other linguistic phenomena that are syntactic in nature in English but they are encoded morphologically in Hungarian. For instance, causation and modality are expressed by derivative suffixes and so is passive (although the passive voice is rare in modern Hungarian): e.g. *csináltathatjátok* (make-CAUS-MODAL-2PL-OBJ) “you can have it made”.

Another peculiarity of Hungarian is that the third person singular present tense indicative form of the copula is phonologically empty, i.e. there are apparently verbless sentences in Hungarian: *A ház nagy* (the house big) “The house is big”. However, in other tenses or moods, the copula is present as in *A ház nagy lesz* (the house big will.be) “The house will be big”.

According to these facts, a Hungarian syntactic parser must rely much more on morphological analysis than e.g. an English one since in Hungarian it is morphemes that mostly encode morphosyntactic information.

3 Morphological Coding Systems for Hungarian

There are three widely used morphological coding systems for Hungarian: Humor, MSD and KR and they make use of different tagsets. The coding system Humor is based on unification, which means that stems and morphemes are assigned features that allow or prohibit their attachment to other morphemes. One word form can contain only morphemes the features of which are not contradictory (Prószéky and Tihanyi, 1993).

[1] Some Hungarian grammars and morphological coding systems treat some rare suffixes as derivational suffixes while others treat them as case suffixes; see e.g. Farkas et al. (2010).

The MSD morphological coding system was developed for a bunch of languages including Hungarian (Erjavec, 2004). Within the codes the first position determines the part-of-speech while other positions offer other types of linguistic information (e.g. in the case of verbs, the type, mood, tense, number and person are provided).

The KR coding system was developed with respect to the morphology of the Hungarian language, however, its basic syntax is language-independent (Trón et al., 2006b). Linguistic information is encoded in hierarchical attribute value matrices: there are default values (e.g singular or 3rd person) and only those that differ from these manifest in the code.

Recently, there has been a successful attempt to harmonize the linguistic principles behind the coding systems MSD and KR (Farkas et al., 2010). The harmonization of Hungarian morphological coding systems was necessary due to the following reasons. *morphdb.hu* is one of the most widely used morphological databases for Hungarian, which makes use of the KR morphological annotation system (Trón et al., 2006a). However, the only manually POS-tagged corpus, the Szeged Corpus (Alexin et al., 2003) is annotated with MSD codes. The two coding systems are not compatible, which entails that if we want to exploit both resources in a statistical language parser (POS tagger, constituency parser, dependency parser etc.), we have to fall back to conversion rules, which leads to the loss of information. In order to avoid this, the two coding systems (MSD and KR) were harmonized and their basic principles were also made compatible. When harmonizing the two coding systems, the following principle was observed: morphological codes should include only those types of information that are useful for later processing (syntax, applications). For instance, in the case of derived verbs, only those pieces of derivational information are explicitly marked that are expressed with syntactic tools in other languages. Recall the example of *csináltathatjátok* (make-CAUS-MODAL-2PL-OBJ) “you can have it made”, where the lemma is *csinál* “make”, the derivational suffixes *-tat* and *-hat* denote causativity and modality, respectively, and the morphological code of the word form includes information on causativity and modality as well. However, no derivational information is marked in the case of the denominal verb *kezel* “treat, handle”, which is derived from *kéz* “hand”, since this information is irrelevant from a syntactic point of view.

4 Related Work

There have been some solutions implemented for the tokenization and morphological analysis of Hungarian texts, which we briefly summarize now.

For tokenizing Hungarian texts, we are aware of the MtSeg segmentation tool developed in the framework of the MULTEXT project (Ide and Véronis, 1994), which was later adapted to Hungarian with the help of specific lists and lexicons (of abbreviations). In addition, the *huntoken* tool also segments Hungarian texts into sentences and tokens and is widely used in many language processing applications (Halácsy et al., 2004).

One of the first morphological analyzer developed for Hungarian was Humor (Prószéky and Tihanyi, 1993). However, the tool is not freely available and is not open source. On the other hand, *hunmorph* is an open source tool, which can be used for lemmatization, morphological analysis and spellchecking in various languages including Hungarian (Trón et al., 2005).

As for Hungarian POS-tagging, *hunpos* was developed on the basis of *hunmorph* (Halácsy et al., 2006). It is based on a Hidden Markov Model, is also free to use and is an open source tool. There is also a POS-tagger based on the morphological analyzer Humor (Prószéky and Tihanyi, 1993), which is enhanced by statistical information gathered from the Hungarian National Corpus (Váradi, 2002). Recently, PurePOS has been implemented (Orosz and Novák, 2012), which is an open source morphological tagger based on a Hidden Markov Model.

Although there are a handful of morphological taggers for Hungarian, their performances are not directly comparable since they rely on different coding systems. However, the harmonized morphology (see Section 3) enable us to build a morphological parser, which is now integrated into `magyarlanc` and the output of which is in total harmony with the Szeged Corpus.

Besides being the first morphological tool that makes use of the harmonized morphological coding system – thus enables the training and evaluation on a large manually annotated corpus –, the most novel feature of *magyarlanc* is that to the best of our knowledge, it contains the first dependency parser adapted to Hungarian.

5 The System

magyarlanc consists of a sentence splitter and a tokenizer, a morphological analyzer and POS-tagger and a dependency parser. In the following, these modules will be presented.

5.1 Sentence Splitting and Tokenization

The first step of text processing is to split the text into sentences, for which we applied the sentence splitter built in MorphAdorner, a language toolkit developed at Northwestern University[2]. Its dictionary was extended with specific Hungarian abbreviations, which end in a dot but they do not signal the end of the sentence, e.g. *kft.* "ltd." or *szül.* "born" and the abbreviations of months. As a second step, tokens within the sentence are identified, which is carried out by the tokenizer module of MorphAdorner. During tokenization, special emphasis is paid to abbreviations consisting of double letters (in Hungarian spelling, some sounds are denoted by a combination of letters, e.g. *cs* denotes the palatal voiceless affricate [tʃ]).

5.2 Morphological Analysis

Lemmatization and morphological analysis is carried out by a morphological analyser based on the lexical resource *morphdb.hu* (Trón et al., 2006a). Originally, the analyzer yields KR morphological codes but they are then converted to the harmonized MSD-style codes (see Section 3). As a result of the morphological analysis, pairs of lemmas and morphological codes are provided for each word. For instance, for the word *egyed* entity / eat-2SG-IMP-OBJ / one-2SGPOSS "entity" / "you should eat" / "your one" we get the following analyses:

egyed@Nn-sn
eszik@Vmmp2s—y
egy@Mc-snd—-s2

where the lemma and the morphological code are separated by an @ sign.

5.3 POS-tagging

POS-tagging is executed by a modified version of the Stanford POS-tagger (Toutanova et al., 2003), which is based on a Maximum Entropy classifier and makes use of the possible tags provided by the morphological analysis (see above). The POS-tagger was trained on the Szeged Corpus, a manually POS-tagged corpus of 1.2 million words (Csendes et al., 2005). For training, we applied only a reduced set of the original MSD-codes, however, at the end of the analysis, full MSD-codes are provided, which are in accordance with the harmonized Hungarian morphology (Farkas et al., 2010). The reduction of POS-codes was necessary as discriminative POS-taggers are not prepared to deal with thousands of different POS-codes. The reduced tagset consisted of only about 60 elements, which proved to be manageable for the POS-tagger.

When reducing the original tagset, we followed the main principle of preserving an unambiguous mapping between the output of the POS-tagger using a reduced tagset on the one hand and the original (full) MSD tagset on the other hand. For instance, a noun ending in the *-nak/-nek* suffix may be in the genitive or in the dative case, thus the MSD codes Nc-sd (a singular noun in the dative) and Nc-sg (a singular noun in the genitive) will be reduced in a different way. However, the codes Nc-sd and Nc-sd---s3 will be reduced to the same form since there is no such Hungarian lemma that would have the same word form for a dative singular and a dative singular with a third person singular possessor (and thus, the POS-tagger would not have to choose between these possibilities).

As default, MSD codes of nouns, adjectives, numerals and pronouns are reduced to the main part of speech (i.e. the first element of the MSD code). Their forms in dative and genitive, however, coincide that is why in these cases the reduced codes also preserve the case of the noun (e.g. Nd, Ng). Essive and superessive forms of nominals may also coincide, e.g. *szépen* nice-ESS or nice-SUP "nicely" or "on a nice one". In such cases, the reduced codes preserve the case as well, e.g. Ap. The form of nouns with a third person singular possessor may coincide with the non-possessive form of the noun, e.g. *Ajkán* Ajka-SUP (a town in Hungary) or lip-SUP "in Ajka" or "on his lip" and here the reduced codes also differ from each other. An inflected form of a third person singular possessive form of a noun with front vowels may coincide with the inflected possessed form of the same noun, e.g. *énekét* song-3SGPOSS-ACC or song-POSS-ACC "his song" or "that of his song",

[2] http://morphadorner.northwestern.edu/

Feature	N	V	V	A	P	T	R	R	S	C	M	I	I	X	Y	Z	O	O
SubPOS	●	●	●	●	●	●	●	l	●	●	●		o				●	e/d/n
Num	●	●	●	●	●			●			●						●	●
Cas	●			●	●						●						●	●
NumP	●			●	●						●						●	●
PerP	●			●	●						●						●	●
NumPd	●			●	●						●						●	●
Mood		●	n															
Tense		●																
Per		●	●		●			●										
Def		●																
Deg				●			●	●										
Clitic																		
Form										●	●							
Coord										●								
Type																		●

Table 1: Relevant features for each part of speech and subtypes of parts of speech: type – SubPOS, number – Num, case – Cas, number of possessor – NumP, person of possessor – PerP, number of possessed – NumPd, mood/form – Mood, tense – Tense, person – Per, definiteness – Def, degree – Deg, clitic – Clitic, form – Form, type of coordination – Coord, subtype – Type.

having the reduced codes `Ns` and `Nz`, respectively. In addition, reduced codes for pronouns belonging to the most important subclasses also preserve their types: `Pe` for personal pronouns, `Pq` for interrogative pronouns and `Pr` for relative pronouns. Fractions also preserve their types (`Mf`).

The default reduced code of a verb is simply `V` and codes of auxiliaries are reduced to `Va`. The present conditional first and second person plural forms of verbs coincide in the objective and subjective conjugation thus the codes of the objective forms are reduced to `Vcp` (e.g. *olvasnánk* read-COND-1PL or read-COND-1PL-OBJ "we would read (an indefinite object)" or "we would read (a definite object)"). For certain verbs, the first person singular forms coincide in subjective and objective conjugation and thus the objective forms are reduced to `Vip` (e.g. *iszom* drink-1SG or drink-1SG-OBJ "I drink (an indefinite object)" or "I drink (a definite object)"). The present conditional first person singular subjective form and the present conditional third person plural objective form of verbs with front vowels also coincide, hence the third person plural form is reduced to `V3p` (e.g. *ennék* eat-COND-1SG or eat-COND-3PL-OBJ "I would eat (an indefinite object)" or "they would eat (a definite object)"). The subjective and objective forms of past indicative first person singular verbs coincide, thus the MSD code of the objective forms is reduced to `Vy` (e.g. *osztottam* divide-PAST-1SG or divide-PAST-1SG-OBJ "I divided (an indefinite object)" or "I divided (a definite object)"). The codes of imperative verbs are reduced to `Vm`. In certain cases the past tense of a verb coincides with the present tense of another verb, thus the MSD codes of present tense verbs for which none of the previous rules hold are reduced to `Vp` (e.g. *ért* (understand) or (reach-PAST-3SG) "understand" or "reached").

MSD codes of adverbs are reduced to `R` by default, however, the most important subtypes of adverbs preserve their types: `Rp` for preverbs, `Rq` for interrogative adverbs, `Rr` for relative adverbs and `Rl` for personal pronominal adverbs. The reduced code of articles is `T`. In the case of conjunctions, postpositions, interjections, abbreviations and misspelled or unknown words, the original MSD code functions as the reduced code as well.

Table 1 shows the relevant features for each part of speech. It is also noted in the table if a specific subclass of a given part of speech has different features than the main part of speech, e.g. not all the grammatical features are relevant for infinitives that are relevant for main verbs.

5.4 Syntactic Parsing

There are two mainstream approaches to syntactic parsing: the one based on constituency grammar and the other one based on dependency grammar. Dependency parsers are believed to be especially useful for parsing languages with free word order such as Hungarian since these parsers are able to connect grammatically related words that are not adjacent.

Farkas et al. (2012) made the first experiments on applying state-of-the-art dependency parsers to Hungarian. Since their results indicated that the Bohnet parser (Bohnet, 2010) was the most efficient on Hungarian dependency parsing, we integrated this parser into `magyarlanc`. The applied model was trained on the Szeged Dependency Treebank, which consists of 82,000 sentences, is manually POS-tagged and contains manually annotated dependency parses for each sentence (Vincze et al., 2010).

Multiword named entities (e.g. *Coca Cola Ltd.*) and multiword numbers (e.g. *42 million*) are treated in a special way. We consider the last word as the head because the last word of multiword units gets inflected in Hungarian and all the previous elements are attached to the succeeding word, i.e. the penultimate word is attached to the last word, the antepenultimate word to the penultimate one etc. with an NE relation for named entities and a NUM relation for numbers.

In the verbless clauses the Szeged Dependency Treebank introduces virtual nodes. This solution means that a similar tree structure is ascribed to the same sentence in the present third person singular / plural and all the other tenses / persons (see Figure 2). A further argument for the use of a virtual node is that the virtual node is always present at the syntactic level since it is overt in all the other forms, tenses and moods of the verb. Seeker et al. (2012) experimented with several methods for inserting virtual nodes into the verbless clauses. Although their results indicate that this issue still requires further investigation, in `magyarlanc`, we follow their complex label approach, which means that children of a virtual node are assigned a complex dependency label (e.g. ROOT-VAN-SUBJ), referring to the fact that the specific node is the subject of a virtual node (here VAN) which is itself not present in the sentence but functions as the root. Figure 2 shows variations of a sentence in the past tense and in the present tense with a virtual node and with complex dependency labels.

In order to represent the dependency parses of the sentences visually, we also integrated the `whatswrong`[3] visualizer into the system.

5.5 The Output of the Toolkit

As an input, `magyarlanc` requires a raw text in a txt format. The linguistic processing can be used in three possible modes. First, it is only tokenization and POS-tagging that is carried out. Second, dependency parsing also takes place beside the above-mentioned two processing steps. Third, it is only morphological analysis that is carried out.

Borpancsolókra	borpancsoló	Nn-ps
,	,	,
zajongókra	zajongó	Nn-ps
és	és	Ccsw
állatkínzókra	állatkínzó	Nn-ps
nagyon	nagyon	Rx
számítanak	számít	Vmip3p—n
.	.	.

Table 2: POS-tagging of the sentence *Borpancsolókra, zajongókra és állatkínzókra nagyon számítanak.* “They heavily count on wine forgers, noise makers and animal torturers.”

The output file produced by `magyarlanc` has the following structure. One line corresponds to one token and sentences are separated by an empty line. When there is no dependency parsing carried out, the first column contains the word form, the second one contains the lemma and the third one contains the MSD code. A sample of the output is shown in Table 2.

When there is also dependency parsing, the first column contains the identifier of the word within the sentence, the second column contains the word form, the third one the lemma, the fourth one the MSD code, the fifth one the part of speech, the sixth one the morphological features, the seventh one the identifier of the parent node, and finally the eighth one contains the dependency label. Table 3 shows a sample output of a sentence parsed both morphologically and syntactically.

Figure 3 shows a sample dependency graph visualized by the `whatswrong` tool. The dependency parse of the sentence is denoted by arrows and the coarse-grained morphological analysis can also be found under the word forms.

6 Results

In order to evaluate the performance of `magyarlanc`, we experimented both with POS-tagging and dependency parsing. For this purpose, we made use of the Szeged Dependency Treebank (Vincze et al., 2010). Sentences of the treebank were randomly divided into training and test sets in a ratio of 80:20%, respectively. Below, we show

[3] `https://code.google.com/p/whatswrong/`

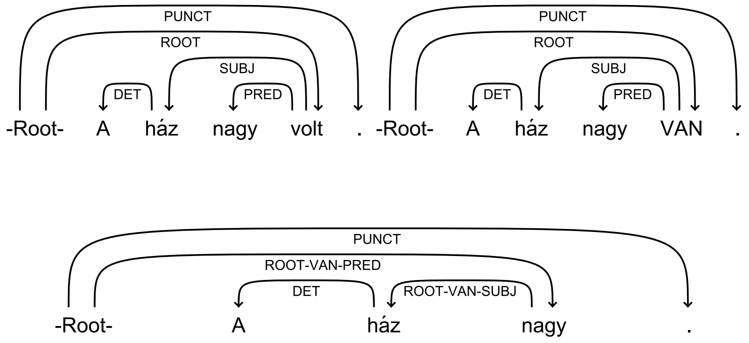

Figure 2: Dependency graphs with overt and covert virtual nodes of the sentences *A ház nagy (volt).* "The house is/was big."

1	Az	az	Tf	T	SubPOS=f	2	DET
2	elnök	elnök	Nn-sn	N	SubPOS=n—Num=s|Cas=n| NumP=none|PerP=none|NumPd=none	3	SUBJ
3	megígérte	megígér	Vmis3s—y	V	SubPOS=m|Mood=i|Tense=s|Per=3|Num=s|Def=y	0	ROOT
4	,	,	,	,	_	3	PUNCT
5	az	az	Tf	T	SubPOS=f	7	DET
6	észlelt	észlelt	Afp-sn	A	SubPOS=f|Deg=p|Num=s|Cas=n| NumP=none|PerP=none|NumPd=none	7	ATT
7	hibákat	hiba	Nn-pa	N	SubPOS=n|Num=p|Cas=a| NumP=none|PerP=none|NumPd=none	14	OBJ
8	a	a	Tf	T	SubPOS=f	9	DET
9	szövetség	szövetség	Nn-sn	N	SubPOS=n|Num=s|Cas=n| NumP=none|PerP=none|NumPd=none	10	ATT
10	vezetése	vezetés	Nn-sn—s3	N	SubPOS=n|Num=s|Cas=n| NumP=s|PerP=3|NumPd=none	14	SUBJ
11	45	45	Mc-snd	M	SubPOS=c|Num=s|Cas=n|Form=d| NumP=none|PerP=none|NumPd=none	12	ATT
12	napon	nap	Nn-sp	N	SubPOS=n|Num=s|Cas=p| NumP=none|PerP=none|NumPd=none	13	OBL
13	belül	belül	St	S	SubPOS=t	14	TLOCY
14	kijavítja	kijavít	Vmip3s—y	V	SubPOS=m|Mood=i|Tense=p|Per=3|Num=s|Def=y	3	ATT
15	.	.	.	.	_	0	PUNCT

Table 3: Morphological and dependency analysis of the sentence *Az elnök megígérte, az észlelt hibákat a szövetség vezetése 45 napon belül kijavítja.* "The president promised that the leadership of the federation would correct the recognized errors within 45 days."

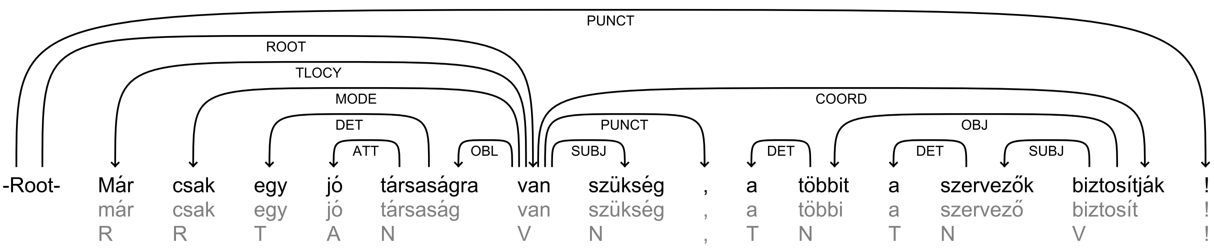
PUNCT
ROOT
TLOCY
MODE
COORD
DET
PUNCT
OBJ
ATT
OBL
SUBJ
DET
DET
SUBJ
-Root- Már csak egy jó társaságra van szükség , a többit a szervezők biztosítják !
már csak egy jó társaság van szükség , a többi a szervező biztosít !
R R T A N V N , T N T N V !

Figure 3: Dependency graph of the sentence *Már csak egy jó társaságra van szükség, a többit a szervezők biztosítják*! "Now you just need a good company, everything else will be provided by the organizers."

POS-tagging	96.33%
Dependency parsing (LAS)	91.42%
Dependency parsing (ULA)	93.22%

Table 4: Results achieved by `magyarlanc`.

and discuss the results of our experiments.

6.1 Results on POS-tagging

In order to determine the efficiency of POS-tagging, we applied an accuracy score. An analysis was considered correct if both the lemma and the deep morphological information (i.e. the part of speech and all the morphological features) of the token were correct. In this way, *magyarlanc* achieved an accuracy of 96.33%.

6.2 Results on Dependency Parsing

For the evaluation of dependency parsing, we applied the metrics Labeled Attachment Score (LAS) and Unlabeled Attachment Score (ULA). In the case of LAS, it is both the parent node and the dependency label that must be the same as the gold standard while in the case of ULA, it is only the parent node that counts (i.e. a wrong dependency label does not yield an error). `magyarlanc` obtained the scores of 91.42% (LAS) and 93.22% (ULA) on the test set.

6.3 Speed of Linguistic Processing

We also tested how fast `magyarlanc` can parse texts. For this purpose, we selected *Stars of Eger*, a historical novel written by Géza Gárdonyi. Running the whole processing chain from segmentation till dependency parsing, 1000 sentences are analyzed per minute by using 1 GB RAM, running on a single thread. If just segmentation and POS-tagging are performed, it results in an analysis of 3000 sentences per minute.

7 Conclusions

In this paper, we presented `magyarlanc`, a natural language toolkit developed for the linguistic preprocessing – segmentation, morphological analysis, POS-tagging and dependency parsing – of Hungarian texts. The main novelties of the tool are the usage of the harmonized morphological coding system of Hungarian and the integration of a dependency parser, which makes it unique among NLP tools developed for Hungarian. It is also data-driven as every module was systematically trained and evaluated on the Szeged Corpus and Szeged Dependency Treebank. The system is implemented in JAVA, hence it can be used on all kinds of platforms. `magyarlanc` is freely available for research purposes at `http://www.inf.u-szeged.hu/rgai/magyarlanc`.

Acknowledgments

This work was supported in part by the European Union and the European Social Fund through the project FuturICT.hu (grant no.: TÁMOP-4.2.2.C-11/1/KONV-2012-0013). Richárd Farkas was partially funded by the Hungarian National Excellence Program (TÁMOP 4.2.4.A/2-11-1-2012-0001).

References

Zoltán Alexin, János Csirik, Tibor Gyimóthy, Károly Bibok, Csaba Hatvani, Gábor Prószéky, and László Tihanyi. 2003. Annotated Hungarian National Corpus. In *Proceedings of the EACL*, pages 53–56.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 89–97.

Dóra Csendes, János Csirik, Tibor Gyimóthy, and András Kocsor. 2005. The Szeged Treebank. In *TSD*, pages 123–131.

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

Tomaš Erjavec. 2004. *MULTEXT-East morphosyntactic specifications. Version 3.*

Richárd Farkas, Dániel Szeredi, Dániel Varga, and Veronika Vincze. 2010. MSD-KR harmonizáció a Szeged Treebank 2.5-ben [Harmonizing MSD and KR codes in the Szeged Treebank 2.5]. In *VII. Magyar Számítógépes Nyelvészeti Konferencia*, pages 349–353.

Richárd Farkas, Veronika Vincze, and Helmut Schmid. 2012. Dependency Parsing of Hungarian: Baseline Results and Challenges. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 55–65, Avignon, France, April. Association for Computational Linguistics.

Péter Halácsy, András Kornai, László Németh, András Rung, István Szakadát, and Viktor Trón. 2004. Creating open language resources for Hungarian. In *Proceedings of the 4th International Conference on Language Resources and Evaluation.*

Péter Halácsy, András Kornai, and Csaba Oravecz. 2006. HunPos - an open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 209–212, Prague, Czech Republic. Association for Computational Linguistics.

Nancy Ide and Jean Véronis. 1994. MULTEXT: Multilingual Text Tools and Corpora. In *Proceedings of the 15th conference on Computational linguistics*, pages 588–592.

György Orosz and Attila Novák. 2012. PurePos – an open source morphological disambiguator. In *Proceedings of the 9th International Workshop on Natural Language Processing and Cognitive Science*.

Gábor Prószéky and László Tihanyi. 1993. Humor: High-speed unification morphology and its applications for agglutinative languages. *La tribune des industries de la langue*, 10:28–29.

Wolfgang Seeker, Richárd Farkas, Bernd Bohnet, Helmut Schmid, and Jonas Kuhn. 2012. Data-driven dependency parsing with empty heads. In *Proceedings of COLING 2012: Posters*, pages 1081–1090, Mumbai, India, December. The COLING 2012 Organizing Committee.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 173–180.

Viktor Trón, László Németh, Péter Halácsy, András Kornai, György Gyepesi, and Dániel Varga. 2005. Hunmorph: open source word analysis. In *Proceedings of ACL*.

Viktor Trón, Péter Halácsy, Péter Rebrus, András Rung, Eszter Simon, and Péter Vajda. 2006a. Morphdb.hu: Hungarian lexical database and morphological grammar. In *Proceedings of 5th International Conference on Language Resources and Evaluation (LREC '06)*.

Viktor Trón, Péter Halácsy, Péter Rebrus, András Rung, Eszter Simon, and Péter Vajda. 2006b. The annotation system of HunMorph. Technical report, The Media Research center of the Budapest University of Technology and Economics.

Tamás Váradi. 2002. The Hungarian National Corpus. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-2002)*, pages 385–389, Las Palmas de Gran Canaria. European Language Resources Association.

Veronika Vincze, Dóra Szauter, Attila Almási, György Móra, Zoltán Alexin, and János Csirik. 2010. Hungarian Dependency Treebank. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10)*.