DOI:10.15774/PPKE.ITK.2015.007

# Hybrid algorithms for preprocessing agglutinative languages and less-resourced domains effectively

*Doctor of Philosophy Dissertation*


**György Orosz**

Roska Tamás Doctoral School of Sciences and Technology

Pázmány Péter Catholic University

Supervisor:

**Gábor Prószéky, DSc**

Budapest, 2015

DOI:10.15774/PPKE.ITK.2015.007

To my beloved wife Jucus and my family.

DOI:10.15774/PPKE.ITK.2015.007

# Acknowledgements

*"My help will come from the Lord, who made heaven and earth."*

– Psalms 121:2

First of all, I would like to say thank you to my scientific advisor, Gábor Prószéky for guiding and supporting me over the years.

I am also thankful to Attila Novák for the fruitful conversations and his useful advice. I would like to give thanks to Nóra Wenszky as well for polishing my English and helping me to refine this study. However, I am solely responsible for all the mistakes, ambiguities and omissions that might have remained in the text.

Conversations, lunches and the always cheerful coffee breaks with my colleagues are greatly acknowledged. Thanks to László Laki, Borbála Siklósi, Balázs Indig, Kinga Mátyus for collaborating in numerous valuable studies. I am also thankful to members of room 314 István Endrédy, Győző Yang Zijian, Bálint Sass, Márton Miháltz, András Simonyi and Károly Varasdi for the friendly and intellectual atmosphere.

I am also grateful to the Pázmány Péter Catholic University and the MTA-PPKE Hungarian Language Technology Research Group, where I spent my PhD years. Thanks are due to current and former leaders Tamás Roska, Judit Nyékyné Gaizler, Péter Szolgay giving me the opportunity to conduct my research. I would like to give special thanks to Katalin Hubay and Lívia Adorján for organizing our conference trips.

Work covered in this dissertation was supported partly by the TÁMOP 4.2.1.B – 11/2/KMR-2011–0002 and 4.2.2/B – 10/1–2010–0014 projects.

Most importantly, I am thankful to my family. I cannot express enough thanks to my loving wife Jucus for tolerating my absence and encouraging me over the years. I am grateful to my parents and brother Tomi for their continuous support during my studies.

DOI:10.15774/PPKE.ITK.2015.007

# Abstract

This thesis deals with text processing applications examining methods suitable for less-resourced and agglutinative languages, thus presenting accurate preprocessing algorithms.

The first part of this study describes morphological tagging algorithms which can compute both the morpho-syntactic tags and lemmata of words accurately. A tool (called PurePos) was developed that was shown to produce precise annotations for Hungarian texts and also to serve as a good base for rule-based domain adaptation scenarios. Besides, we present a methodology for combining tagger systems raising the overall accuracy of Hungarian annotation systems.

Next, an application of the presented tagger is described that aims to produce morphological annotation for speech transcripts, and thus, the first morphological disambiguation tool for spoken Hungarian is introduced. Following this, a method is described which utilizes the adapted PurePos system for estimating morpho-syntactic complexity of Hungarian speech transcripts automatically.

The third part of the study deals with the preprocessing of electronic health records. On the one hand, a hybrid algorithm is presented for segmenting clinical texts into words and sentences accurately. On the other hand, domain-specific enhancements of PurePos are described showing that the resulting tagger has satisfactory performance on noisy medical records.

Finally, the main results of this study are summarized by presenting the author's theses. Further on, applications of the methods presented are listed which aims less-resourced languages.

DOI:10.15774/PPKE.ITK.2015.007

# List of Figures

DOI:10.15774/PPKE.ITK.2015.007

# List of Tables

DOI:10.15774/PPKE.ITK.2015.007

DOI:10.15774/PPKE.ITK.2015.007

# Abbreviations

**CHILDES** Child Language Data Exchange System

**CRF** Conditional Random Fields

**ERR** Error rate reduction

**HMM** Hidden Markov model

**HUKILC** Hungarian Kindergarten Language Corpus

**MA** Morphological analyzer

**maxent** Maximum entropy

**ML** Machine learning

**MLE** Maximum likelihood estimation

**MLU** Mean length of utterance

**MLUm** Mean length of utterance in morphemes

**MLUw** Mean length of utterance in words

**MRE** Mean Relative Error

**NLP** Natural language processing

**OOV** Out of vocabulary

**PoS** Part-of-speech

**SB** Sentence boundary

**SBD** Sentence boundary detection

DOI:10.15774/PPKE.ITK.2015.007

# Table of Contents

DOI:10.15774/PPKE.ITK.2015.007

DOI:10.15774/PPKE.ITK.2015.007

DOI:10.15774/PPKE.ITK.2015.007

# 1

# Introduction

## 1.1 Preprocessing in natural language technology

Natural language technology is present in our everyday life helping interactions between humans and computers. As such, it is a field of computer science and linguistics, which involves understanding and generating of human (natural) languages. Concerning text processing, which is a major part of language technology, several structural levels can be identified [22]:

**Text segmentation:** basic units of texts are separated, thus token and sentence boundaries are recognized (referred as tokenization sentence boundary detection (SBD)).

**Morphological parsing:** structural units of words are identified (morphological analysis), then tokens are unambiguously classified by their morpho-syntactic behavior (part-of-speech tagging).

**Syntactic parsing:** sentences are broken down into building blocks regarding their form, function or syntactic relation to each other.

**Semantic analysis** methods deal with the *meaning* of texts.

DOI:10.15774/PPKE.ITK.2015.007



Practical applications often build parsing chains, pipelining such components one after another. Two preprocessing steps are indispensable for most of the cases. Since words and sentences are the basic units of text mining applications, segmentation must be performed first. Beside this, lemmata and part-of-speech (PoS) labels of words are also necessary components of such systems, thus morphological parsing should be carried out next.

Moving on, such pipelined architectures may easily result in erroneous output, since error propagation is often a notable phenomenon. Obviously, the more accurate preprocessing modules are employed, the better analyses are yielded. Therefore, the high precision of such methods is crucial.

## 1.2 A solved problem?

Text segmentation is composed of two parts: tokenization and sentence boundary identification. The first one brakes texts into meaningful elements (called tokens) usually utilizing pattern matching methods. Next, sentence boundaries are often recognized by applying linguistic rules or using machine learning algorithms (cf. [23]). In most of the cases, these solutions are fine-tuned for a specific task, hence resulting in accurate tools, i.e. the problem is considered to be solved. However, these algorithms are often language and domain specific, thus numerous scenarios exist (such as the case of noisy texts) on which current approaches fail (see [24]).

Having identified the tokens themselves, the PoS tags of words are assigned. In practice, these solutions mostly build on data-driven algorithms requiring large amount of training data. As a result, such approaches are restricted by the corpus they model. Further on, most of the tagging algorithms target English first, thus ignoring serious problems caused by languages with rich morphology. For instance, agglutinative languages (such as Hungarian) have rich inflection systems. Words are formed joining affixes to the lemma, thus affecting their morpho-syntactic behavior. In that way, such languages need much larger (morpho-syntactic) tag-sets compared to English [25].

DOI:10.15774/PPKE.ITK.2015.007



Furthermore, lemmatization of words cannot be carried out using simple suffix-stripping methods. This means, disambiguating among part-of-speech labels becomes insufficient (see e.g. [26]), full morphological tagging algorithms are required that assign complete morpho-syntactic tags and compute lemmata as well.

All in all, language technology needs preprocessing methods which handle morphologically rich languages efficiently and perform well on less-resourced scenarios at the same time.

## 1.3 Aims of the study

The aim of this study is twofold. Firstly, morphological tagging algorithms are investigated which can handle agglutinative languages and is applicable for domain adaptation scenarios effectively. Secondly, methods suitable for a less-resourced domain are examined.

First, we were interested in **how existing methods can be applied for the full morphological tagging of agglutinative languages yet remaining suitable for domain adaptation tasks**. Chapter 2 considers many aspects of this question. Section 2.2.2 focuses on the full disambiguation problem, in particular on the question of **how one can create a morphological tagging architecture that is accurate on agglutinative languages and also flexible enough to be used in rule-based domain adaptation tasks**. Further on, this section also investigates **how a method can be created which computes roots of words (either seen or unseen previously by the algorithm) effectively**. On the one hand, an efficient lemmatizer was developed which integrates a morphological analyzer (MA) and employs several stochastic models as well. On the other hand, an efficient tagging tool (PurePos) is designed that is customizable for diverse domains. Our system was tested on a general Hungarian corpus showing its state-of-the-art accuracy. In addition, hybrid components of the tool were also examined through an annotation task showing their conduciveness.

DOI:10.15774/PPKE.ITK.2015.007



Following this, Section 2.3 examines **how one can improve full morphological taggers through system combination to raise the overall annotation quality**. We developed an architecture for combining morphological taggers for agglutinative languages, which improves tagging quality significantly.

Beside tagging methods, their applications also played a central role in this study. We were interested in **creating a tagger tool for speech transcripts which can help linguists in their research**. Chapter 3 presents adaptation methods resulting in the first morphological tagging chain for spoken Hungarian. Following this, an application of this system is described which estimates morpho-syntactic complexity of speech transcripts of children automatically.

The third part of the dissertation (Chapter 4) deals with problems of Hungarian electronic health records. In particular, Section 4.2 investigates **how one can develop a text segmentation algorithm which can handle imperfect sentence and word boundaries in Hungarian medical texts**. Our contribution in this field is twofold. First, it was shown that all the available tools fail on segmenting such texts. Next, an accurate methodology was proposed identifying sentence and token boundaries precisely.

Following this, Section 4.3 looks into the questions **what the main pitfalls of morphological taggers are which target noisy clinical texts** and **how PurePos can be adapted for tagging medical texts properly**. This part introduces a detailed error analysis of the tool showing that abbreviations and out-of-vocabulary (OOV) words cause most of the errors. In addition, domain-specific adaptation techniques are presented improving the annotation quality significantly.

## 1.4 Methods of investigation

In the course of our work, diverse corpora were used. First, the Szeged Corpus [27] was employed for developing and evaluating general tagging methods. Further on, these algorithms were tested on Old and Middle Hungarian [12] texts as well. Next, methods for speech transcripts were analyzed on the HUKILC corpus [2].

DOI:10.15774/PPKE.ITK.2015.007



Beside existing ones, two new corpora were created manually from electronic health records. These texts enabled us to design algorithms for the clinical domain. Concerning their usage, texts were usually split into training, development and test sets.

As regards methods used, most of our work resulted in hybrid solutions. On the one hand, we built on symbolic morphological analyzers and rule-based (pattern matching) components. On the other hand, stochastic and machine learning algorithms were heavily utilized as well.

Morphological analyzers played a central role in our study, since their usage is inevitable for morphologically complex languages. In most of the cases we employed (adapted versions [12, 28, 16]) of Humor [29, 30, 31] but the MA of `magyarlanc` [32] was used as well.

As regards machine learning algorithms, tagging experiments were based on hidden Markov models [33, 34]. Our approach built on two well-known tools which are Brant's TnT [35] and HunPos [36] from Halácsy et al. Besides, other common methods such as $n$-gram modeling, suffix-tries and general interpolation techniques were utilized as well. Further on, the proposed combination scheme applied instance-based learning [37] implemented in the Weka toolkit [38].

Beside supervised learning, unsupervised techniques were employed as well. Identification of sentences was performed using the collocation extractions measure of Dunning [39]. In fact, we based on the study of Kiss and Strunk [40], which employs scaling factors for the $\log \lambda$ ratio.

The effectiveness of algorithms was measured calculating standard metrics. The performance of taggers were computed with accuracy as counting correct annotations of tokens and sentences. However, if the corpus investigated contained a considerable amount of punctuation marks, they were not involved in the computation. For significance tests, we used the paired Wilcoxon signed rank test as implemented in the SciPy toolkit [41]. Next, the improvement of taggers was examined calculating relative error rate reduction.

DOI:10.15774/PPKE.ITK.2015.007



Simple classification scenarios were evaluated computing precision, recall and F-score for each class. Furthermore, overall accuracy values were provided as well. Finally, numeric scores were compared with mean relative error [42] and Pearson's correlation coefficient [42].

DOI:10.15774/PPKE.ITK.2015.007

# 2

# Full morphological tagging methods

## 2.1 Motivation

Is morphological tagging really a solved task? Although several attempts have been made to develop tagging algorithms since the 1960's (e.g. [43, 44]), those were focusing mainly on English word classes. Further on, such approaches usually concentrated only on increasing PoS taggers' accuracy on news text, while e.g. problems of other domain are still barely touched. In addition, recently there has been an increasing interest on processing texts in less-resourced languages, which are morphologically rich (cf. [45, 46, 47, 48]). Most of them are highly inflectional or agglutinative, posing new challenges to researchers. This study gives an account of the morphological tagging of agglutinating languages by investigating the case of Hungarian.

First of all, a remarkable difficulty for tagging agglutinative languages is data sparseness. If we compare (cf. [49]) languages like Hungarian or Finnish with English in terms of the coverage of vocabularies by a corpus of a given size, we find that although there are a lot more different word forms in the corpus, these still cover a much smaller percentage of possible word forms of the lemmata in the corpus than in the case of English. On the one hand, a 10 million word English corpus has less than 100,000 different word forms, while a corpus of the same size for Finnish or Hungarian contains well over 800,000. On the other hand, while an open class English word has maximally

DOI:10.15774/PPKE.ITK.2015.007



4–6 different inflected forms, it has several hundred or thousand different productively suffixed forms in agglutinative languages. Moreover, there are much more disparate possible morpho-syntactic tags for such languages than in English (several thousand vs. a few dozen). Thus, the problem is threefold:

1. an overwhelming majority of possible word forms of lemmata occurring in the corpus is totally absent,
2. words in the corpus have much fewer occurrences, and
3. there are also much fewer examples of tag sequences (what is more, several possible tags may not occur in the corpus at all).

Another issue for morphologically rich languages is that labeling words with only their part-of-speech tag is usually insufficient. Firstly, complex morpho-syntactic features carried by the inflectional morphemes can not be represented by tag-sets having only a hundred different labels. Secondly, morpho-syntactic tagging is still just a subtask of full morphological disambiguation. In addition to a full morpho-syntactic tag, lemmata of words also need to be identified. Although several studies have revealed that dictionary- or rule-based lemmatization methods yield acceptable results for morphologically not very rich languages like English [50, 51], ambiguity is present in the task for highly inflectional and agglutinative languages [52, 53, 54]. Yet, most of the taggers available only concentrate on the tag but not the lemma, thus doing just half of the job.

Looking into the details, there are annotation schemes (e.g. MSD codes of the Szeged Corpus [27]) which eliminate the ambiguity of the lemmatization task by the tag-set they use. This means that roots of words should be calculated undoubtedly from the morphological categories assigned. Nevertheless, lemma computing still can be important problem in these cases. Tagger tools operating without any prior morphological knowledge still have to figure out how to derive lemmata from morphological labels. This means that they have to infer knowledge about the inner structures of words. What is more, the same issue also holds when a MA is used,

DOI:10.15774/PPKE.ITK.2015.007



but the word is unknown to that analyzer system and is not seen in the training data. For handling these cases, morphological disambiguator tools should employ guessing methods dealing effectively with such out-of-vocabulary words.

Moving on, lemmatization is a more important problem when the annotation scheme does not restrict the lemmatization task as seen above. As regards Hungarian, one can reveal notable ambiguity investigating the Szeged Corpus [27] with the Humor analyzer [29, 30, 31]. First of all, more than 16% of words are ambiguous by their lemmata, furthermore, if we aggregate morphological analyses by their morpho-syntactic label, 4% of the tokens still have more than one roots. An example is a class of verbs that end in *-ik* in their third person singular present tense indicative, which is the customary lexical form (i.e. the lemma) of verbs in Hungarian. Further on, another class of verbs has no suffix in their lemma. The two paradigms differ only in the form of the lemma, so inflected forms can belong to the paradigm of either an *-ik* final or an non-*ik* final verb and many verbs. E.g. *internetezem* 'I am using the internet' can have two roots for the same morpho-syntactic tag: *internetez* and *internetezik* 'to use the internet'. Another example is the class of verbs which third person singular past causative case overlap with the singular third person past form. For example *festette* 'he painted it/he made it to be painted' has two possible roots for the same tag: *festet* 'he makes someone to paint' and *fest* 'he paints'.

Besides, a further issue is that most of the tagging approaches perform well only when a satisfactory amount of training data is available. In addition, several agglutinative languages and especially their subdomains lack annotated corpora. Concerning Hungarian, even though the Szeged Corpus contains well over 80,000 sentences, there are several important domains (such as the case of biomedical texts) which miss manually annotated documents. Therefore, pure stochastic methods that are trained on this corpus and target other genres may result in low quality annotation.

In this chapter, we present an effective morphological tagging algorithm that has a language independent architecture being capable of annotating sentences with full morpho-syntactic labels and lemmata. The presented method has state-of-the-art

DOI:10.15774/PPKE.ITK.2015.007



performance for tagging Hungarian. Most importantly, it is shown that our tool can be used effectively in resource-scare scenarios, since it yields high quality annotations even when a limited amount of training data is available only. Finally, tagger combination experiments are presented raising further the accuracy of Hungarian morphological tagging.

## 2.2 Hybrid morphological tagging methods

This section surveys related studies first. Following this, a new hybrid morphological tagging algorithm is described detailing its components and architecture. Finally, the presented tool is evaluated through several experiments showing its high performance.

### 2.2.1 Background

First of all, we overview how morphological tagger systems are typically built up. Since there are just a few tools performing the full task, we also review PoS tagging attempts for morphologically rich languages. In addition, previous approaches for Hungarian are introduced as well.

#### 2.2.1.1 Full morphological tagging

There has been insufficient discussion about full morphological tagging in recent studies of natural language technology. The reason behind this is that most of the attempts concentrate on morphologically not very rich languages (such as English), where PoS tagging is generally sufficient, and the ambiguity of the lemmatization task is negligible. Furthermore, there are studies (following English approaches) which ignore lemmatization (such as [55, 56, 57]) even for highly inflectional languages.

Nevertheless, approaches on full morphological tagging can be grouped depending on their relationship to lemmatization.

DOI:10.15774/PPKE.ITK.2015.007



1. First of all, numerous researchers propose a two stage model, where the first phase is responsible for finding full morpho-syntactic tags, while the second one is for identifying lemmata for *(word, tag)* pairs. For instance, Erjavec and Dzeroski decompose the problem [58] by utilizing a trigram tagger first, then applying a decision list lemmatizer. Further on, Agič et al. combine [59] an HMM-based tagger with a data-driven rule-based lemmatizer [26]. Even though such combinations could have error propagation issues, they usually result in well-established accuracy.

2. Another feasible approach is to treat the tagging task as a disambiguation problem. Such methods utilize morphological analyzers to generate annotations candidates, then employ disambiguation methods for selecting correct analyses. These architectures are typical e.g. for Turkish attempts (cf. [53, 60]). A drawback of this approach is that the disambiguation component depends heavily on the language-dependent analyzer used.

3. Finally, the problem can be handled as a unified tagging task. An example is the Morfette system [54]. It employs a joint architecture for tagging words both with their tags and lemmata considering lemmatization as a labeling problem. The tool represents a lemma class as a transformation sequence describing string modifications from the surface form to the root. Further on, Morfette utilizes the maxent framework and employs separate models for each of the subtasks yet using a joint beam search decoder. Another similar method was presented by Laki and Orosz [7] recently. Their system (HuLaPos) merges PoS labels with lemmata transformation sequences to a unified tag, which is then learned by the Moses statistical machine translation framework [61]. Therefore, HuLaPos can translate sentences to sequences of labels. These joint approaches are usually language independent, however, they can either be slow to train or inaccurate due to the increased search space they use.

DOI:10.15774/PPKE.ITK.2015.007



Considering lemmatization (case 1 above), the task can be easily accomplished by utilizing linguistic rules or lemma dictionaries. However, the creation of such resources is time-consuming. Next, a general baseline method is to select the most frequent lemmata for each *(word, tag)* relying on the training data (as in [32]). Despite its simplicity, this method usually results in mediocre precision systems. Further on, employing advanced machine learning (ML) algorithms is also a viable approach. E.g. Plisson et al. apply ripple down rule induction algorithms [51] for learning suffix transformations. Even though they report about good results, their attempt ignores the dependency between tags and lemmata. Next, Jongejan and Dalianis generate decision lists (cf. CST method [26]) for handling morphological changes in affixes. However, their system is optimized for inflecting languages exploring complex changes in word forms.

#### 2.2.1.2 Morpho-syntactic tagging of morphologically rich languages

Next we describe how well-known data-driven PoS tagging methods are applied for morphologically rich languages focusing on issues yielded by the complexity of the morphology. In doing so, only data-driven models are reviewed investigating techniques for managing

1. the increased number of out-of-vocabulary word forms and
2. the large complexity of the tag-set.

While numerous attempts have been published for tagging Polish recently [62, 63, 64, 65], performance of these tools are below the average. Most of these solutions (e.g. [65]) use morphological analyzers to get morpho-syntactic tag candidates to reduce the search space of the decoder used. Further on, tiered tagging is another widely utilized technique [65]. This method resolves complex tags by computing its parts one after another. Considering ML algorithms used, the range of applications is wide. Beside an adaptation of Brill's tagger [64], C4.5 decision trees [63], memory-based learning [66] and CRF models are employed [65] as well.

DOI:10.15774/PPKE.ITK.2015.007



Moving on, the first successful attempt to analyze Czech was published by Hajič and Hladká [55] basing on a discriminative model. Their approach uses a morphological analyzer and builds on individual prediction models for ambiguity classes. Actually, the best results for Czech are obtained using the combination of numerous systems [67]. In their solution, three different data-driven taggers (HMM, maximum entropy and averaged perceptron) and further symbolic components are utilized as well. A MA computes the possible analyses, while the rule-based disambiguator tool removes agrammatical tag sequences.

The flexible architecture of the Stanford tagger [68] also allows the integration of various morphological features thus enabling its usage for morphologically rich languages. An example is the Bulgarian tool [69] (by Georgiev et al.), which uses a morphological lexicon and an extended feature set. Further on, applications of trigram tagging methods [35, 36] have been demonstrated (for example Croatian [59], Slovenian [59] and Icelandic [70]) to be effective as well. These systems achieve high accuracy utilizing large morphological lexicons and decent unknown word guessing algorithms.

Considering agglutinative languages, the usage of finite-state methods is indispensable for handling the huge number of possible wordforms. E.g. Silferberg and Lindén introduce a trigram tagger for Finnish [57] that is based on a weighted finite-state model. As regards Turkish, Daybelge and Cicelki describe a system [71] employing also a finite-state rule-based method. However, most taggers for agglutinative languages use hybrid architectures incorporating morphological analyzers into stochastic learning methods. Examples are the perceptron-based application of Sak et al. [53] and the trigram tagging approach of Dilek et al. [60].

Recent approaches include the results of the Statistical Parsing of Morphologically Rich Languages workshop [45, 46, 47, 48]. First, Le Roux et al. [72] presented an accurate parser for Spanish relying on the morphological annotations of Morfette [54]. Further on, Bengoetxea et al. [73] showed that tagging quality greatly influences the accuracy of parsing Basque. For this, they used a tagger based on hidden Markov models

DOI:10.15774/PPKE.ITK.2015.007



combined with a symbolic component. Next, Maier et al. [74] investigated the effect of the tag-set granularity on processing German. They concluded, that PoS tagging can be performed more accurately using less granular tags, while both the coarse-grained and too fine-grained morpho-syntactic labels decrease the parsing performance.

Most recently, Bohnet et al. have introduced methods [75] for the joint morphological and syntactic analysis of richly inflected languages. Their best solutions involve the usage of morphological analyzers and word clusters, resulting in significant improvements on parsing of all the tested languages. However, their method requires syntactically annotated texts restricting the applicability of the algorithm for less-resourced domains. Further on, Müller et al. [76] improved on CRF-based methods to apply them effectively on the morpho-syntactic tagging of morphologically rich languages. The proposed system uses a coarse-to-fine mapping on tags for speeding-up the training of the underlying discriminative method. In this way, their solution can go beyond general 1$^{st}$-order models thus resulting in increased accuracy. Their best systems utilize complex morpho-syntactic features and outputs of morphological analyzers as well.

To summarize, effective methods for morphologically complex languages rely on either a discriminative or a generative model. Such taggers generally use morphological lexicons or analyzers to handle the large vocabulary of the target language. Further on, tagging of unknown words is a crucial problem being managed by either guessing modules or rich morphological features.

#### 2.2.1.3 The case of Hungarian

As regards Hungarian PoS tagging, the first attempt was performed by Megyesi [77] adapting Brill's transformation-based method [78]. Since she did not use any morphological lexicon, her approach resulted in moderate success. Similarly, Horváth et al. investigated [79] only pure machine learning algorithms (such as C4.5 or instance based learning) resulting in low accuracy systems. Three years later, the first promising approach was presented by Oravecz and Dienes [49] utilizing TnT [35] with a weighted

DOI:10.15774/PPKE.ITK.2015.007



finite-state lexical model. In 2006, Halácsy et al. investigated an augmented maxent model [80] in combination with language specific morphological features and a MA. In that study, the best result was achieved by combining the latter model with a trigram-based tagger. Later, they created the HunPos system [36], which reimplements and extends TnT. Their results (with a morphological lexicon) has been shown to be as efficient as the one of Oravecz and Dienes [49]. Next, Kuba et al. applied boosting and bagging techniques for transformation based learning [81]. Although they managed to reduce the error rate of the baseline tagger, their results lag behind previous approaches. Recently, Zsibrita et al. published `magyarlanc` [32], a natural language processing chain specially designed for Hungarian. They adapted the Stanford tagger [68] in two steps. First, to train the underlying discriminative method effectively, a tag-transformation step is applied reducing the number of morpho-syntactic labels. Next, the tool utilizes a morphological analyzer as well to enhance and speed up the disambiguation process.

Most of the previous approaches concentrated on the morpho-syntactic tagging task, thus there are only three tools which performs lemmatization as well and can be applied to Hungarian.

1. `magyarlanc`, which builds on a maxent model and employs a rule-based lemma guesser system.

2. HuLaPos, which applies machine translation methods for the tagging task.

3. Morfette, which uses two maximum entropy models for jointly decoding both the labels and lemmata.

Although, `magyarlanc` provides accurate annotations, the tool has two weaknesses which inhibit its applications on corpora having different annotation schemata. It does not provide a straightforward way to train and contains built-in annotation-specific components. Further on, machine learning algorithms behind HuLaPos and Morfette need a large amount of training data to produce high accuracy, therefore it can be

DOI:10.15774/PPKE.ITK.2015.007



troublesome to apply them on less-resourced domains. Finally, none of these tools contain components which would allow them to be adjusted for new domains without retraining them.

This chapter introduces a new morphological tagging tool, which uses simple trigram methods and employs a pluggable morphological analyzer component. It differs from existing approaches in having a language and tag-set independent architecture yet producing high accuracy and also being flexible enough to be used in rule-based domain-adaptation scenarios. PurePos is designed to operate on morphologically rich languages and resource-scarce scenarios, providing accurate annotations even when a limited amount of training data is available only. Furthermore, our approach is shown to have very high accuracy on general Hungarian.

### 2.2.2 The full morphological tagging model

The architecture of PurePos (cf. Figure 2.1) is composed of multiple components. The data flow starts from a MA providing word analyses as *(lemma, tag)* pairs. Next, a trigram model is used to select morpho-syntactic labels for words. Then, lemmatization is carried out using both statistical and linguistic components.

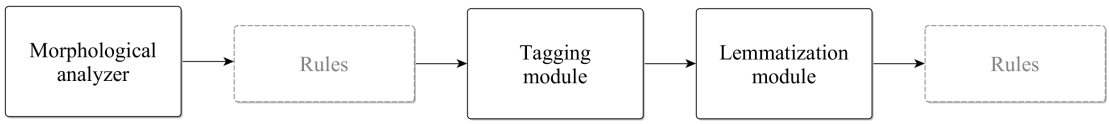


Figure 2.1 The architecture of the proposed method

In the following, we present its components making the morphological tagging effective. Underlying statistical models are introduced first, then we show how symbolic algorithms are incorporated.

DOI:10.15774/PPKE.ITK.2015.007



#### 2.2.2.1 The PoS tagging model

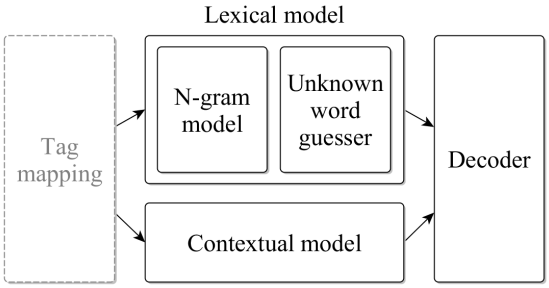


Figure 2.2 Part-of-speech tagging in the proposed system

PurePos builds on HMM-based methods [33, 34] introduced in TnT [35] and HunPos [36], allowing it to be fast, simple and effective at the same time. Our implementation (similarly to HunPos) allows the user to set the tagging order individually for both the contextual ($n_1$) and lexical model ($n_2$). The method presented (see Figure 2.2) selects the best fitting $t_1^m$ morpho-syntactic label sequence for the $m$ long $w_1^m$ sentence using individual contextual and lexical probabilities of tags and words (in the $i$th position):

$$\arg\max_{t_1^m} \prod_{i=1}^{m} P(t_i|t_{i-1}^{i-n_1}) P(w_i|t_{i-1}^{i-n_2}) \tag{2.1}$$

Its contextual model is computed with simple $n$-gram language-modeling techniques (cf. Equation 2.2) employing maximum likelihood estimation (MLE) (see Equations 2.3 and 2.4)[1]. Uni-, bi- and trigram estimates are combined with deleted interpolation thus calculating $\lambda_k$ weights as suggested by Brants [35]. Even though the order of the model is usually set to 3, it is adjustable in practice.

$$P(t_i|t_{i-1}^{i-n_1}) \approx \sum_{k=0}^{n_1-1} \lambda_k \hat{P}(t_i|t_{i-1}^{i-k}) \tag{2.2}$$

$$\hat{P}(t_i|t_{i-1}^{i-k}) = \frac{c(t_i^{i-k})}{c(t_{i-1}^{i-k})} (k > 0) \tag{2.3}$$

[1] Where $N$ denotes the size of the tag-set, while $c(x)$ marks the number of $x$ elements in the training data.

DOI:10.15774/PPKE.ITK.2015.007



$$\hat{P}(t_i) = \frac{c(t_i)}{N}(k=0) \tag{2.4}$$

Next, the lexical model ($P(w_i|t_{i-1}^{i-n_2})$ of our method is composed of two components. The first one handles tokens previously seen in the training data, while the second guesses labels for unknown words. In fact, each subsystem is doubled (as it is in [35, 36]) maintaining separate models for uppercase and lowercase words.

Handling of previously seen words is carried out approximating $P(w_i|t_i^{i-n_2})$ with word-tag co-occurrences:

$$P(w_i|t_i^{i-n_2}) \approx \sum_{k=0}^{n_2-1} \lambda_k \hat{P}(w_i|t_i^{i-k}) \tag{2.5}$$

$\hat{P}(w_i|t_i^{i-k})$ is calculated with maximum likelihood estimation, while deleted interpolation is applied with $\lambda_k$ weights. As in the contextual model, $k$ is set to 2 in applications.

As regards tagging of unknown words, we use – in accordance with Brants – the distribution of rare[2] tokens' tags for estimating their PoS label. Since suffixes are strong predictors for tags in agglutinative languages, we use the last $l$ letters ($\{s_{n-l+1}\dots s_n\}$) for estimating probabilities. Successive abstraction is utilized in our tool as described in [34, 35]. This method calculates the probability of a $t$ tag recursively using suffixes with decreasing lengths:

$$P(t|s_{n-l+1},\dots,l_n) \approx \frac{\hat{P}(t|s_{n-l+1},\dots,s_n) + \theta\hat{P}(t|s_{n-l},\dots,s_n)}{1+\theta} \tag{2.6}$$

$\theta$ parameters are computed utilizing the standard deviation of the maximum likelihood probabilities of all the $k$ tags:

[2]Rare words are considered to be those that occur less than 10 times in the training data.

DOI:10.15774/PPKE.ITK.2015.007



$$\theta = \frac{1}{k-1}\sum_{j=1}^{k}(\hat{P}(t_j) - \overline{P})^2 \tag{2.7}$$

where

$$\overline{P} = \frac{1}{k}\sum_{j=1}^{k}\hat{P}(t_j) \tag{2.8}$$

Finally, MLE is employed for calculating both $\hat{P}(t_j)$ and $\hat{P}(t|s_{n-l+1},\ldots,s_n)$.

Concerning decoding, beam search is utilized, since it can yield multiple tagging sequences at the same time. In that way, the tool is also able to produce tagging scores of sentences (2.9) allowing us to incorporate further components using partly disambiguated word sequences.

$$Score(w_1^m, t_1^m) = \log\prod_{i=1}^{m} P(w_i|t_i,t_{i-1})P(t_i|t_{i-1},t_{i-2})P(l_i|t_i,w_i) \tag{2.9}$$

#### 2.2.2.2 The lemmatization model

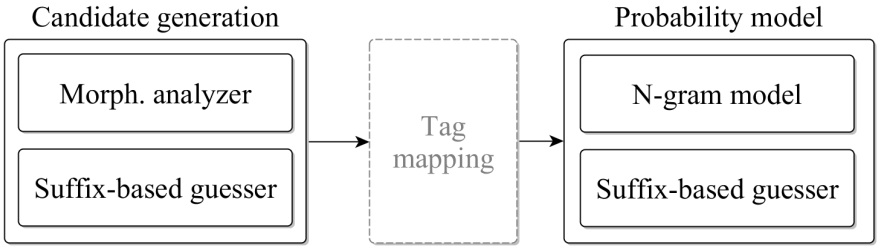


Figure 2.3 The data flow in the lemmatization component

Lemmatization is performed in two steps (cf. Figure 2.3). First, candidates are generated for *(word, morpho-syntactic tag)* pairs. If morphological analyses are available for the current word, their lemmata are used as candidates, otherwise suffix-based guessing is carried out. For this, the guesser (described in Section 2.2.2.1) was extended

DOI:10.15774/PPKE.ITK.2015.007



to handle lemma transformations as well. Combined labels can represent both the morpho-syntactic tag and suffix-transformations for lemmata (for an example see Table 2.1).

Table 2.1 Examples for the combined representation of the tag and lemma

| | | |
|---|---|---|
| **Word** | *házam* 'my houses' | *baglyot* 'owl' |
| **Tag** | N.1sPOS | N.ACC |
| **Lemma** | *ház* 'house' | *bagoly* 'owl' |
| **Transformation** | -2+∅ | -4+*oly* |
| **Combined label** | (N.1sPOS, -2,–) | (N.ACC, -4, *oly*) |

As for picking the right lemma, we utilize a simple scoring model (2.10) that evaluates candidates using their part-of-speech tags:

$$\arg\max_l S(l|t,w) \tag{2.10}$$

This method is based on a twofold estimation of $P(l|t,w)$. On the one hand, a unigram lemma model ($P(l)$) calculates conditional probabilities using relative frequency estimates. On the other hand, reformulation of $P(l|t,w)$ yields another approximation method:

$$P(l|t,w) = \frac{P(l,t|w)}{P(t|w)} \tag{2.11}$$

Substituting this formula to (2.10), $P(t|w)$ becomes a constant which can be omitted. In that way, we can estimate $P(l,t|w)$ employing only the lemma guesser. Finally, models are aggregated in a unified ($S$) score:

$$S(l|w,t) = P(l)^{\lambda_1} P(l,t|w)^{\lambda_2} \tag{2.12}$$

The idea of computing $\lambda_{1,2}$ parameters is similar to that seen for the PoS $n$-gram models. However, instead of using positive weights, negative scores are stored for the better model. $\lambda_k$ is calculated iterating over words of the training data (cf. Algorithm 1):

DOI:10.15774/PPKE.ITK.2015.007



**Algorithm 1** Calculating parameters of the lemmatization model

```
for all (word, tag, lemma) do
    candidates ← generateLemmaCandidates(word, tag)
    maxUnigramProb ← getMaxProb(candidates, word, tag, unigramModel)
    maxSuffixProb ← getMaxProb(candidates, word, tag, suffixModel)
    actUnigramProb ← getProb(word, tag, lemma, unigramModel)
    actSuffixProb ← getProb(word, tag, lemma, suffixModel)
    unigramProbDistance ← maxUnigramProb − actUnigramProb
    suffixProbDistance ← maxSuffixProb − actSuffixProb
    if unigramProbDistance > suffixProbDistance then
        λ2 ← λ2 + unigramProbDistance − suffixProbDistance
    else
        λ1 ← λ1 + suffixProbDistance − unigramProbDistance
    end if
end for
normalize(λ1, λ2)
```

1. first, both components return the best roots for each *(word, tag)* pair,
2. then probability estimates for the gold standard lemma are computed,
3. next, (absolute) error rates of the models are calculated ,
4. finally, the best model's weight is decreased[3].

After these steps, $\lambda_k$ parameters are normalized.

#### 2.2.2.3 Hybridization

Although, the framework proposed builds on an existing PoS tagging algorithm, it is extended with a new lemmatization model and is modified to fit agglutinative languages such as Hungarian. Hybridization steps listed below show the differences between PurePos and its predecessors [35, 36].

[3] Since probability estimates are between 0 and 1, decreasing a weight gives higher values.

DOI:10.15774/PPKE.ITK.2015.007



**Morphological analyzer**

First of all, a morphological analyzer is utilized throughout the whole process, therefore probability estimation is performed for valid[4] analyses only.

**Linguistic rules**

Next, the presented architecture allows rule-based components to modify the analyses of the MA, in that way, bad candidates can be filtered out. Furthermore, lexical probability scores of analyses can be also given to PurePos, which are then used as context-dependent local distribution functions.

**Unseen tags**

In contrast to TnT or HunPos, our system is able to handle unseen tags[5] properly. On the one hand, if a token has only one analysis not seen before, that one gets selected with 1 lexical probability. Further on, estimation of forthcoming tags is performed using a lower level (unigram) model in this case. On the other hand, the system can also calculate lexical and contextual scores for any tag previously not seen. This can be performed mapping latter tags to known ones using regular expressions.[6]

***k*-best output**

Finally, our method decodes tags using beam search. One can generate partly disambiguated sentences being apt for linguistic post-processing. Further on, this facility allows the usage of advanced machine learning techniques resulting in more accurate parsing algorithms.

[4] Valid analyses for a word are those which are proposed by the MA.
[5] Morpho-syntactic labels which are not seen in the training data.
[6] For a complete example see Section 2.2.3.3.

DOI:10.15774/PPKE.ITK.2015.007



### 2.2.3 Experiments

#### 2.2.3.1 Tagging general Hungarian

First, PurePos is evaluated on Hungarian texts. We used the Szeged Corpus [27] (SZC) for our experiments, since it is the only Hungarian resource which is manually annotated and is freely available. It contains general texts from six genres being annotated with detailed (MSD) morpho-syntactic tags [25] and lemmata.

On the one hand, we used the original corpus (version 2.3[7]). On the other hand, a variant of the SZC was employed as well that is tagged with the analyses of Humor [29, 30, 31]. Using both of them, we could evaluate our algorithm with two different morphological annotation schemata. They were split in 8:2 ratio (randomly) for training and testing (as in Table 2.2) purposes. Since the two corpora are not aligned which each other (the transcribed one contains fewer sentences), results obtained on the two datasets are not directly comparable.

Table 2.2 Dimensions of the corpora used

| | MSD tag-set | | Humor tag-set | |
|---|---|---|---|---|
| | Training set | Test set | Training set | Test set |
| Tokens | 1,232,384 | 254,880 | 980,225 | 214,123 |
| Sentences | 68,321 | 13,778 | 56,792 | 14,198 |
| Distinct tags | 1,032 | 716 | 983 | 656 |

As a morphological analyzer is an integral part of our method, we tested the tool with two different modules. The first setting utilized the MSD tagged corpus and an analyzer extracted from `magyarlanc`, while the second one applied Humor on the transcribed corpus.

Evaluation was carried out measuring the overall accuracy of full annotations (i.e *(morpho-syntactic tag, lemma)* pairs). For significance tests, we used the Wilcoxon matched-pairs signed-rank test at the 95% confidence level dividing the test set into 100

[7]The MA that provides MSD annotations is only compatible with this corpus variation.

DOI:10.15774/PPKE.ITK.2015.007



data pairs. Sentence-based accuracies were also provided in some cases. The latter metric was computed by considering a sentence to be correct only when all of its tokens are properly tagged.

We compared our results with other morphological tagging tools available for Hungarian[8]. Firstly, taggers providing full morphological annotations such as `magyarlanc`, HuLaPos and Morfette[9] were evaluated. Secondly, we assembled full morphological taggers from available components[10], since two-phase architectures were also shown to be prosperous (e.g. [59, 58]).

Concerning PoS tagging, we used three of the most popular algorithms as baselines. These are the following:

- the trigram tagging method of HunPos,
- averaged perceptron learning and
- the maximum entropy framework of the OpenNLP [82] toolkit.

As regards lemmatization, CST [26] and a simple baseline method (BL) were employed. The latter one assigns the most frequent lemmata to a previously seen *(word, tag)* pairs, otherwise the root is considered to be the word itself.

Beside these components, tag dictionaries were prepared for HunPos, since it can employ such resources. At this point we simulated a setting, where the tagger was only loaded once. Therefore, a large lexicon was prepared for the tool. In that way, analyses of the 100,000 most frequent words of Hungarian were provided to the tagger[11].

[8] Since, our aim was only to compare available tagger methods, not to optimize each of them, external tools were employed with their default settings.

[9] Version 3.5 is used.

[10] PoS taggers are trained using the full morpho-syntactic labels of words.

[11] Frequencies are calculated relying on the results of the Szószablya project [83].

DOI:10.15774/PPKE.ITK.2015.007



Table 2.3 Tagging accuracies of Hungarian taggers on the Szeged Corpus (annotated with MSD labels)

| | PoS tagging | Morph. tagging | |
|---|---|---|---|
| | | Token | Sentence |
| `magyarlanc` | 96.50% | 95.72% | 54.52% |
| Morfette | 96.94% | 92.24% | 38.18% |
| HuLaPos | 96.90% | 95.61% | 54.57% |
| PurePos | 96.99% | 96.27% | 58.06% |
| HunPos + BL | 96.71% | 92.65% | 36.06% |
| HunPos + CST | 96.71% | 91.19% | 35.31% |
| Maxent + BL | 95.63% | 92.21% | 34.82% |
| Maxent + CST | 95.63% | 90.14% | 29.70% |
| Perceptron + BL | 95.19% | 91.16% | 29.42% |
| Perceptron + CST | 95.19% | 89.78% | 27.91% |

Table 2.4 Tagging accuracies of Hungarian taggers on the transcribed Szeged Corpus (annotated with Humor labels)

| | PoS tagging | Morph. tagging | |
|---|---|---|---|
| | | Token | Sentence |
| Morfette | 97.60% | 94.73% | 51.58% |
| HuLaPos | 97.19% | 95.53% | 57.55% |
| PurePos | 98.65% | 98.58% | 81.78% |
| HunPos + BL | 97.41% | 89.93% | 32.07% |
| HunPos + CST | 97.41% | 94.69% | 52.40% |
| Maxent + BL | 94.81% | 88.82% | 28.19% |
| Maxent + CST | 94.81% | 92.33% | 40.10% |
| Perceptron + BL | 95.97% | 88.85% | 29.11% |
| Perceptron + CST | 95.97% | 93.32% | 45.13% |

DOI:10.15774/PPKE.ITK.2015.007



First of all, there are notable discrepancies between the results on the two datasets (cf. Tables 2.3 and 2.4). On the one hand, performance discrepancies can be explained by the morphological analyzers used. These tools have different coverage, thus they affect the results of parsing chains built on them. On the other hand, the two corpora utilize different annotation schemes:

- First, the original corpus contains foreign and misspelled words being tagged with a uniform X tag. Due to the various syntactic behavior of such tokens, their labels could not be estimated using their context or their suffix properly.

- Further on, date expressions and several named entities are tagged with a single MSD code resulting in lemmata composed of more than one words. (An example is *Golden Eye-oztunk* 'we visited the Golden Eye' being lemmatized as *Golden Eye-ozik* 'to visit the Golden Eye'.) Such phenomena could be hard to handle for lemmatizers.

These variations can have a huge impact on morphological disambiguation algorithms. In our case, they decrease the accuracy of MSD-based systems, while allow Humor-based ones to produce better annotation (since the corresponding corpus is free of such phenomena).

In general, results show that the best-performing systems are PurePos, HuLapos, `magyarlanc` and Morfette. Besides, HunPos also achieves high PoS tagging scores, while the other two-stage taggers are far behind state-of-the-art results. As regards learning methods of the OpenNLP toolkit, their performance indicate that they cannot handle such labeling problems precisely. Further on, both of the standalone lemmatizers degrade accuracy. This reduced performance can be due to their design: the baseline method was not prepared for handling unknown words, while CST was originally created for inflectional languages. An interesting difference between lemmatization scores is that the baseline (BL) strategy performs better on the original corpus, while the CST tool gives higher accuracy on the Humor-labeled dataset. A reason behind this phenomena can be that the latter dataset has higher lemma ambiguity (cf. Section 2.1) thus requiring advanced methods.

DOI:10.15774/PPKE.ITK.2015.007



An explanation for the high morpho-syntactic labeling score of PurePos is that it uses morphological analyzers to get analysis candidates. This component can reduce the number of unknown words, thus enabling the system to provide better annotations. While HunPos also uses such resources, it can only handle static lexicons, which limits its accuracy.

As regards `magyarlanc`, it also yields first-class results (95.72% accuracy) on the MSD-tagged corpus. However, its built-in language (and annotation scheme) specific components inhibited its application on the other corpus. Further on, HuLaPos is an interesting outlier. It is based on pure stochastic methods, but it can still achieve high precision. These results can be explained by the larger contexts used by the underlying machine translation framework. Next, Morfette also provides accurate annotations (concerning PoS labels only), however, it has problems with computing the roots of words.

Results show that PurePos provides the most accurate annotations amongst tested tools. Further on, its advance over other systems is statistically significant (Wilcoxon test of paired samples, $p < 0.05$)[12]. In addition, sentence-based accuracies (especially on the Humor-tagged corpus) also confirms the superior performance of our method. These values reveal that most of the tools result in erroneously tagged sentences in more than half of the cases, while the same number for our method is much less (18% on the transcribed corpus and 42% on the original one).

It was shown that the presented algorithm can produce high quality morpho-syntactic annotations handling the huge number of inflected forms of Hungarian. Further on, it can also produce precise root candidates for both previously seen and unseen tokens due to its improved lemma computing method. In that way, PurePos is a suitable tool for morphological tagging of Hungarian.

[12]Pairwise tests were carried out comparing token-level accuracies of PurePos and other competing tools on each corpus.

DOI:10.15774/PPKE.ITK.2015.007



#### 2.2.3.2 Resource-scarce settings

Next, PurePos is compared to other taggers on less-resourced scenarios. For this, we use systems[13] and corpora described in Section 2.2.3.1. To simulate such settings, when just a limited amount of training data is available, we trained all the taggers using a few thousand sentences only (cf. Table 2.5). As an evaluation, learning curves of systems are drawn on both versions of the test set.

Table 2.5 The number of tokens and sentences used for training the taggers for simulating resource-scarce settings

| Sentences | 2,000 | 4,000 | 6,000 | 8,000 | 10,000 |
|---|---|---|---|---|---|
| Tokens (MSD-tagged corpus) | 13,555 | 26,496 | 53,563 | 79,916 | 107,113 |
| Tokens (Humor-tagged corpus) | 20,863 | 41,740 | 82, 964 | 121,026 | 146,816 |

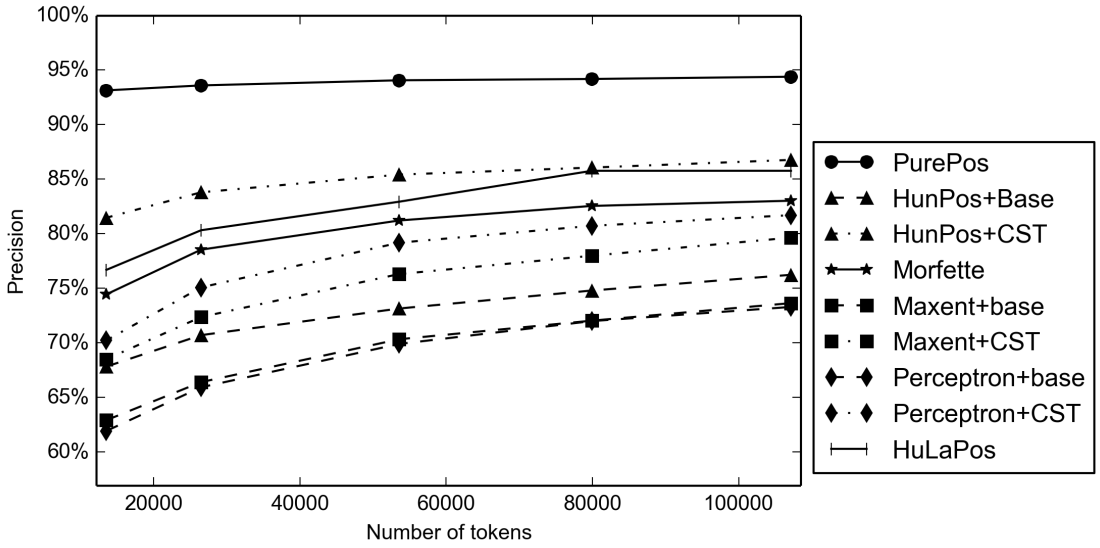


Figure 2.4 Learning curves (regarding token accuracy) of full morphological taggers on the Szeged Corpus (using MSD labels)

[13] Unfortunately, we could not measure the performance of `magyarlanc`, since the current release of the tool cannot be trained.

DOI:10.15774/PPKE.ITK.2015.007



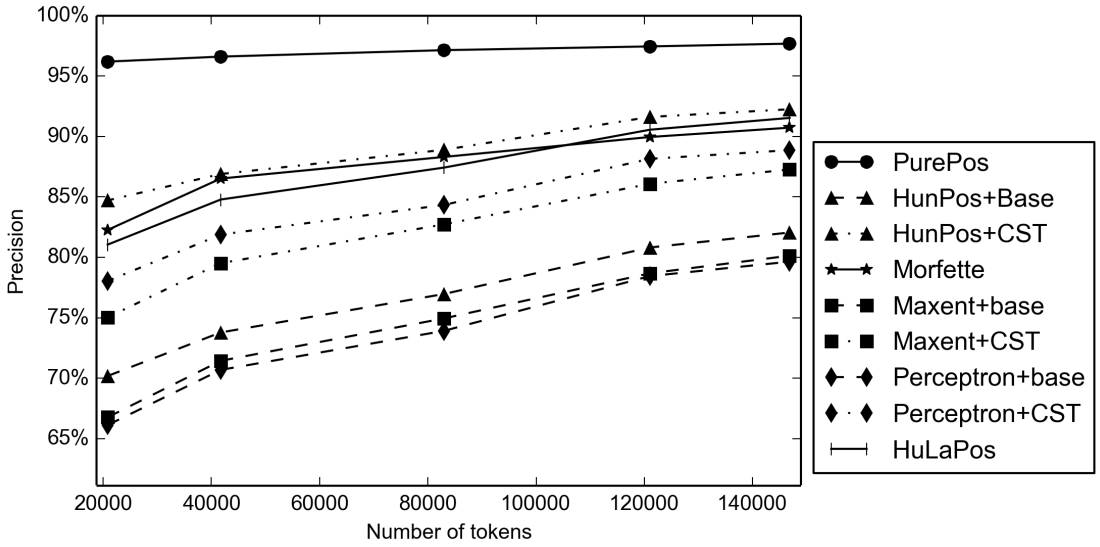


Figure 2.5 Learning curves (regarding token accuracy) of full morphological taggers on the Szeged Corpus (using Humor labels)

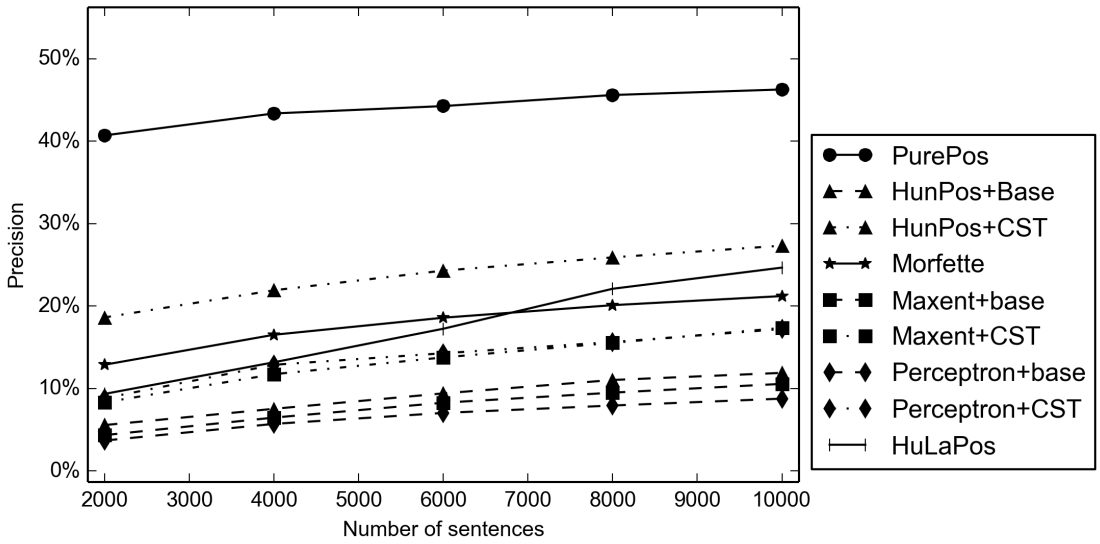


Figure 2.6 Learning curves (regarding sentence accuracy) of full morphological taggers on the Szeged Corpus (using MSD labels)

DOI:10.15774/PPKE.ITK.2015.007



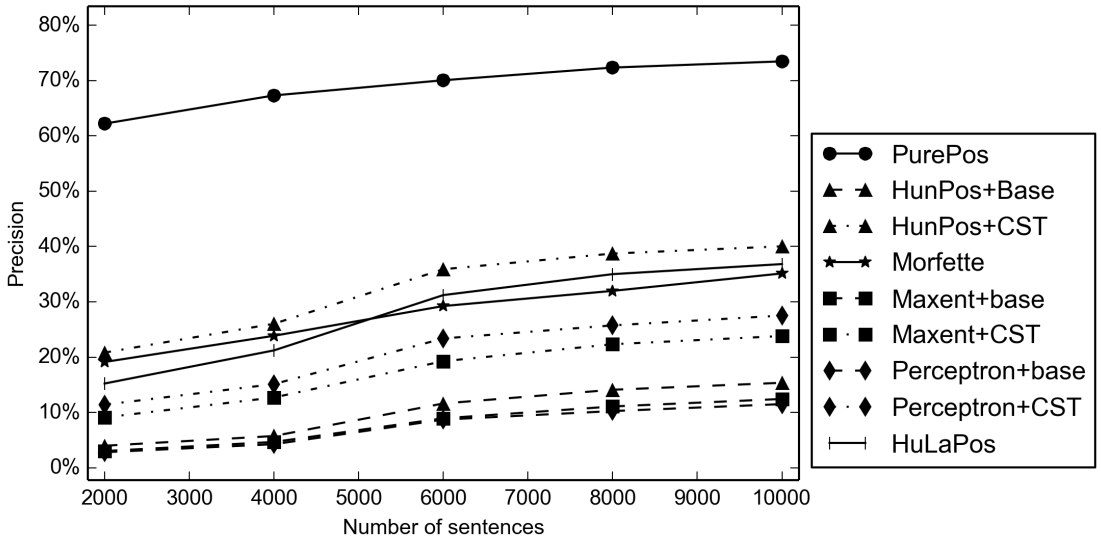


Figure 2.7 Learning curves (regarding sentence accuracy) of full morphological taggers on the Szeged Corpus (using Humor labels)

First, Figures 2.4 and 2.5 present morphological tagging accuracies of systems depending on the number of tokens in the training corpus. These results are in accordance with conclusions of our previous experiments; however, the differences revealed are higher. Further on, the large distance between the accuracy scores of PurePos and other tools confirms the effectiveness of our hybrid approach in less-resourced scenarios.

Additionally, if we compare (cf. Figures 2.6 and 2.7) the sentence-based accuracies of the taggers, the gap between their performance are much more emphasized. For example, having only 2,000 sentences for training (with MSD tags) the proposed algorithm results in 40.71% sentence-level accuracy compared to the second best of 18.62%. The increased performance of our method is in a great part due to two things. On the one hand, the system presented extensively use morphological analyzers, restricting the number of candidate analyses effectively thus providing more accurate analyses for OOV tokens. On the other hand, Markov models are known to perform better in the case of resource-scare scenarios compared to discriminative methods.

DOI:10.15774/PPKE.ITK.2015.007



In brief, we have shown that the architecture of PurePos allows producing accurate annotations when the amount of training data is limited. Therefore, our method could be used for morphological tagging scenarios when there is just a few thousand manually annotated sentences are available.

#### 2.2.3.3 The case of Middle- Old-Hungarian

Next, we present a tagging task showing the effectiveness of all the hybrid components available in PurePos. In a project [28, 12] aiming at the creation of an annotated corpus of Middle Hungarian texts, an adapted version of the Hungarian Humor morphological analyzer [28] was used[14]. This tool was originally made to annotate contemporary Hungarian, but the grammar and lexicon were modified to handle morphological constructions that existed in Middle Hungarian but have since disappeared from the language. In the experiments described here, we used a manually disambiguated portion of this corpus. The tokens were labeled using a rich variant of the Humor tag-set having cardinality over a thousand.

Table 2.6 Number of clauses and tokens in the Old and Middle Hungarian corpus

| | Training | Development | Test |
|---|---|---|---|
| Documents | 140 | 20 | 30 |
| Clauses | 12,355 | 2,731 | 2,484 |
| Tokens | 59,926 | 12,656 | 11,763 |

The corpus was split into three parts (see Table 2.6) for the experiments. The tagger was trained on the biggest one, adaptation methods were developed on a separate development subcorpus, while final evaluation was done on the test set. We used accuracy as an evaluation metric, but unambiguous punctuation tokens were *not* taken into account (in contrast to how taggers are evaluated in general). They are ignored because the corpus contains a relatively large amount of punctuation marks which

[14]The adaptation of Humor and the annotation were done by Attila Novák and Nóra Wenszky. The author's contribution is the enhancement of the morphological tagging chain.

DOI:10.15774/PPKE.ITK.2015.007



would distort the comparison. Methods were evaluated in two ways: full morphological disambiguation accuracies were calculated for tokens and they were also computed to obtain clause-level accuracy values. In addition, error rate reduction (ERR) (2.13) is calculated measuring the percentage of mistakes ($E$) of a baseline tagger ($b$) that are corrected by an enhanced method ($n$).

$$\mathrm{ERR}(b,n) = \frac{E(b) - E(n)}{E(b)} \tag{2.13}$$

We used the improved trigram-based algorithm derived from HunPos and implemented in PurePos (PP) as a baseline PoS tagger. This basic chain is enhanced step-by-step investigating the impact of each component. First, the MA and the new lemmatization method is analyzed on the development set (cf. Table 2.7).

Table 2.7 Baseline disambiguation accuracies on the development set. BL is the baseline unigram lemmatizer, while CL is đthe proposed one. PPM and PP both denote the PurePos tagger, however the first uses a morphological analyzer.

| | Tokens | Clauses |
|---|---|---|
| PP + BL | 88.99% | 55.58% |
| PPM + BL | 97.22% | 84.85% |
| PP + CL | 92.14% | 65.40% |
| PPM + CL | 97.58% | 86.48% |

On the one hand, we compare the PoS tagging method of PurePos with (PPM) and without the morphological analyzer (PP). On the other hand, the simple unigram-based (BL) lemmatizer (cf. Section 2.2.2.1) is evaluated against the proposed one (CL). First, it was found that the usage of a morphological component is indispensable. Next, results show that the proposed algorithm yields a significant error rate reduction compared to the baseline. This improvement is even more notable (28.42% ERR) when a dedicated morphological analyzer is not used.

DOI:10.15774/PPKE.ITK.2015.007



Below, several experiments are presented to exhaust hybrid facilities of PurePos, thus yielding a more accurate tagger. To that end, the development set was utilized to analyze common error types and to develop hypotheses.

**Mapping of tags** In contrast to other Hungarian annotation projects, the tag-set of the historical corpus distinguishes verb forms that have a verbal prefix from those that do not, because this is a distinction important for researchers interested in syntax.[15] This practically doubles the number of verb tags[16], which results in data sparseness problems for the tagger. In the case of a never encountered label having a verbal prefix marking, one can calculate probability estimates for that tag by mapping it to one without a verbal prefix. This solution is viable, since the distribution of prefixed and non-prefixed verbs largely overlap. Applying this enhancement (TM), we could increase the accuracy of the system on the development set (to 86.53% clause level accuracy) notably.

**Preprocessing** Another point of improvement is to filter analyses of Humor (FI). Exploiting the development set, a preprocessing script was set up which has five simple rules. Three of them catches the tagging of frequent phrases such as *az a* 'that' in which *az* must be a pronoun. Further on, two domain specific lexicons were employed to correct the erroneous annotation of proper names that coincide with frequent common nouns or adjectives. Using these correction rules the overall performance on the development set was further raised to 86.77% clause accuracy.

***k*-best output** The *k*-best output of the tagger can either be used as a representation to apply upstream grammatical filters to or as candidates for alternative input to higher levels of processing. Five-best output for our test corpus has yielded an upper limit for attainable clause accuracy of 94.32% (on the development set). While it is not directly comparable with the ones above, this feature could e.g. be used by syntactic parsers.

[15] Hungarian verbal prefixes or particles behave similarly to separable verbal prefixes in most Germanic languages: they usually form a single orthographic word with the verb they modify, however, they are separated in certain syntactic constructions.

[16] 320 different verb tags occur in the corpus excluding verb prefix vs. no verb prefix distinction. This is just a fraction of the theoretically possible tags.

DOI:10.15774/PPKE.ITK.2015.007



Table 2.8 Disambiguation accuracies of the hybrid tool on the test set. TM is the tag mapping approach, while FI denotes the rule-based preprocessing.

| | Token | Clauses |
|---|---|---|
| Baseline | 89.47% | 55.07% |
| PurePos | 96.48% | 80.95% |
| + TM | 96.51% | 81.17% |
| + FI | 96.60% | 81.55% |
| + all | 96.63% | 81.77% |
| + all with *k*-best | 98.66% | 92.30% |

Enhancements are validated evaluating them on the test set. Data in Table 2.8 show that each linguistic component improves the overall chain significantly[17]. Further on, using the 5-best output sequence of the tagger one can further improve the accuracy of the tool. Golden tags and lemmata are available for 92.30% of the clauses and for 98.66% of the tokens between the top five annotation sequence.

We have shown that one can further increase the tagging accuracy by employing hybrid facilities of PurePos. First, rules were employed filtering our erroneous analysis candidates, then unseen tags were mapped to previously seen ones successfully. Finally, we have shown that the 5-best output contains significantly more golden annotations.

## 2.3 Combination of morphological taggers

Although high accuracy tagging tools are generally available, sometimes their performance is not satisfactory and shall be increased further. In cases where very high annotation quality is required, variance of tools should be considered. Disparate methods can result in different sorts of errors, therefore their combination can yield better algorithms. Even though this idea is not new, being present in both machine learning and NLP literature, there has not been much work done on in this field for morphologically rich languages.

[17] We used the Wilcoxon matched-pairs signed-rank test at $p < 0.05$

DOI:10.15774/PPKE.ITK.2015.007



In this section, the case of agglutinative languages is investigated by experimenting with Hungarian morphological disambiguator tools. First, we give a brief overview about PoS tagger combination methods. Next, discrepancies of two taggers are presented allowing us to create a new combination architecture. Finally, we evaluate the proposed method by measuring its improvement over baseline tools used.

### 2.3.1 Background

For reviewing previous attempts on tagger combination approaches we rely on a recent study of Enríquez et al. [84]. Their work not just investigates existing methods in detail, but also evaluates them for various NLP tasks (involving PoS tagging). Introducing the problem, the authors seek answers for the following question: "What does it take for the combination to be successful?". They conclude (referring to Hansen and Salamon [85]) that there are two fundamental requirements for the success. Classifiers employed must make different sorts of errors, and at the same time these algorithms should be more accurate than a random classifier.

Kuncheva [86] and Enríquez et al. [84] differentiate four levels of combination strategies. Firstly, the meta-classifier is a point of decision, since it greatly effects how categories suggested by the base classifiers are merged. Secondly, one can decide on algorithms used as base classifiers. Thirdly, when feature vectors are utilized to represent examples, their variation can also impact the final result. Finally, different datasets can be utilized to generate diverse taggers.

Regarding Enríquez et al. [84] and Xu et al. [87], combination algorithms can be also distinguished by their input. Such methods can either receive only the PoS labels from the input taggers or it can get a more general output. A ranked list of labels – where part-of-speech categories are sorted by their confidence level – contains much more information. Furthermore, lists containing relevancy scores give the most background knowledge for combination methods.

DOI:10.15774/PPKE.ITK.2015.007



Several ensemble strategies have been applied to PoS tagging in the literature, including voting, bagging, boosting, stacking (cf. [88, 89, 81, 90, 91, 92]) or even using rules for aggregating outputs of input taggers [93]. Two of the most influential studies are the ones which presented by Brill and Wu [88] and Halteren et al. [89]. The former work presents the first attempts of combining English taggers. The authors propose a memory-based meta-learning scheme which employs contextual and lexical clues. In their experiments, the solution where the top-level learner always selects the output of one of the embedded taggers outperformed the more general scheme that allowed the output differ from either of the proposed tags. The comprehensive study of Halteren et al. [89] compares several ensemble methods on three different corpora, showing that stacking methods can be used efficiently to train top-level classifiers for an optimal utilization of the corpus. They found a scheme performing best characterized as generalized voting.

A system of different architecture is presented by Hajič et al. [94]: in contrast to the parallel and hierarchical architecture of the systems above, it employs a serial combination of annotators starting with a rule-based morphological analyzer, followed by constraint-based filters feeding statistical taggers at the end of the chain.

We extend the approach of and Brill and Wu [88] and Halteren et al [89] by adapting their method to the morphological tagging of an agglutinative language. Our ensemble method builds on only the abstract output of input taggers without knowing their rank or score. We use stacking with features adapted to Hungarian, furthermore, language-specific symbolic components are also utilized in our system. To produce a full morphological tagger, the underlying architecture is modified to generate lemmata candidates as well.

### 2.3.2 Discrepancies of taggers

Evaluating taggers on general Hungarian can show that two of the best performing tools (our new method and HuLaPos) significantly diverge by the errors they made. In our evaluation scheme, a detailed analysis of errors was carried out first, aiming to reveal

DOI:10.15774/PPKE.ITK.2015.007



their possible combined performance. For this, we utilized the Humor-tagged Szeged Corpus (described in 2.2.3.1). We kept the training sentences (80% of the corpus), but the rest was split into two parts. The first half was employed for development purposes, while the second one was set apart for the final evaluation (cf. Table 2.9).

Table 2.9 Number of sentences and tokens used for training, tuning and evaluating combination algorithms

| | Tokens | Sentences |
|---|---|---|
| Training set | 980,225 | 56,792 |
| Development set | 105,779 | 7,099 |
| Test set | 108,344 | 7,099 |

First of all, we could not compare word class error rates one-by-one to reveal differences of taggers, since the cardinality of the tag-set is over 1,000. Further on, we could neither rely on Brill's well-known formula (cf. [88]), as it gives hard-to-interpret unlimited negative values when there is a considerable amount of overlap between the errors investigated. Therefore, a new metric, called Own Error Rate (OER), was introduced to measure the relatedness of the taggers' errors. We used the formula

$$\mathrm{OER}(A,B) = \frac{\#\text{errors of } A \text{ only}}{\#\text{errors of either } A \text{ or } B} \tag{2.14}$$

for calculating the percentage of tagger $A$ being wrong but $B$ being correct in proportion of all errors made by either $A$ or $B$.

Table 2.10 Error analysis of PurePos (PP) and HuLaPos (HLP) on the development set

| | Tagging | Lemmatization | Full disambig. |
|---|---|---|---|
| Agreement rate | 97.60% | 98.02% | 96.92% |
| They are right when they agree | 99.30% | 99.85% | 99.29% |
| One is right when they disagree | 97.53% | 98.89% | 97.14% |
| OER(PP, HLP) | 22.41% | 11.66% | 21.16% |
| OER(HLP, PP) | 53.58% | 80.21% | 58.24% |

DOI:10.15774/PPKE.ITK.2015.007



To begin, we investigated the agreement of tools on the development set. As Table 2.10 shows:

1. they agree on the full annotation in most of the cases,
2. matching tags and lemmata are almost always right,
3. one of them (frequently) knows the correct annotation even when their guesses do not match.

Secondly, own error rates indicate that even though HuLaPos performs worse than PurePos, the errors are fairly balanced between them.

Table 2.11 Accuracy of the oracle and the baseline systems on the development set

| | Tagging | Lemmatization | Full disambig. |
|---|---|---|---|
| PurePos | 98.57% | 99.58% | 98.43% |
| HuLaPos | 97.61% | 98.11% | 97.03% |
| Oracle | 99.26% | 99.83% | 99.22% |

Finally, the theoretical maximum performance of the combination (marked as oracle) is presented in Table 2.11. Assuming a hypothetical oracle always selecting the correct *(tag, lemma)* pairs from the tools' suggestions, the accuracy of the better tagger can be further increased eliminating 72.73% of PurePos' errors.

### 2.3.3 Improving PurePos with HuLaPos

To utilize combination through cross-validation, the training set was split into 5 equal-sized parts. Level-0 taggers (PurePos and HuLaPos) were trained 5 times using the 4/5 of the corpus while the rest of the sentences were annotated by both taggers in each round. The union of these automatically annotated parts were used to train the (level-1) metalearners. Furthermore, this technique allowed us to utilize all the training data in each level, yet separating the two phases of the training process.

DOI:10.15774/PPKE.ITK.2015.007



Concerning the question of choosing a level-1 learner we followed Witten et al. [42] for investigating only "relatively global, smooth" algorithms. We utilized [18] the naïve Bayes (NB) classifier [95] and instance-based (IB) learners [37]. The latter in addition to be simple, had been previously shown to perform well in similar combination tasks. Another important decision was to apply metalearners only in cases of disagreement, since the tools' agreement rate was high.

There are at least two parameters which must be set for IB learners. First, a distance function needs to be selected, then the number of neighborhooding events has to be restricted. In that way, we opted on using Manhattan distance and decided to rely only on the single closest item.

Moving on, Hungarian has a tag-set with a cardinality of over a thousand and an almost unlimited vocabulary. Therefore, we applied meta-algorithms choosing the tagger but not the tag.

As regards features, we relied on the set proposed by Brill and Wu [88], since it had been shown (cf. [89]) to be simple but powerful. It (FS1 in Table 2.12) consists of several lexical properties such as

1. the word to be tagged,
2. immediate neighbours of the token,
3. tags suggested for the corresponding word,
4. tags suggested for neighbouring tokens.

[18] C4.5 decision tree algorithm was involved in our experiments, but it was unable to handle the large amount of feature data used.

DOI:10.15774/PPKE.ITK.2015.007



Table 2.12 Feature sets used in the combination experiments

| Feature set | Base features | Additional features |
|---|---|---|
| FS1 | Brill-Wu | — |
| FS2 | FS1 | whether the word contains a full stop or hyphen |
| FS3 | FS1 | use at most 5-character suffixes instead of the word form |
| FS4 | FS2, FS3 | — |
| FS5 | FS1 | guessed tags for the second word both to the right and left |
| FS6 | FS4 | use at most 10-character suffixes instead of the word form |

First, we examined how these attributes can be extended systematically (see Table 2.12) to fit languages with a productive morphology. Since wordforms in Hungarian are composed of a lemma and numerous affixes, longer suffixes features are utilized to handle data sparseness issues. Further on, wider context were also employed to manage the free word order nature of the language.

Performing the experiments, we used the WEKA machine learning toolkit [38]. Improvements were measured on PoS tagging, lemmatization, as well as on the full annotation scenario.

Table 2.13 Error rate reduction of combination algorithms on the development set. IB is the instance based learning algorithm, while NB denotes naïve Bayes.

| Task: | Tagging | | Lemmatization | | Full annotation | |
|---|---|---|---|---|---|---|
| Feature set | NB | IB | NB | IB | NB | IB |
| FS1 | 19.03% | 24.65% | -6.21% | 22.24% | 5.06% | 22.89% |
| FS2 | 18.91% | 24.82% | -0.80% | 23.85% | 4.95% | 23.16% |
| FS3 | 21.04% | 27.60% | 0.80% | 26.65% | 18.42% | 25.31% |
| FS4 | 20.92% | 27.90% | 4.01% | 26.65% | 18.96% | 25.20% |
| FS5 | 16.37% | 17.55% | -19.24% | 16.03% | -0.70% | 18.47% |
| FS6 | 19.27% | 27.30% | -17.03% | 26.85% | 16.16% | 25.79% |

DOI:10.15774/PPKE.ITK.2015.007



Table 2.13 shows error rate reduction scores of different systems compared to PurePos. These results reveal that naïve Bayes classifier (NB) performs significantly worse than instance-based learners (IB) even when using seemingly independent features. Further on, lemma combination turned out to be an insoluble task for that classifier. Improvements show that word shape features (FS2) always help on tagging, while increased contexts (FS5) are not as powerful. An interesting outcome of combining PoS taggers was that the word to be tagged was not necessary amongst the features (see FS4 and FS6 at Table 2.13). However, utilization of longer suffixes boosts the performance. In addition, they are also beneficial in cases where lemmatization is part of the task.

Now we turn on experiments yielding the best combination architecture for full morphological annotation.

#### 2.3.3.1 Combination of morphological taggers

A simple combination structure is to treat annotations as atomic units letting the metalearner choose the output of one of the baselines (cf. Figure 2.8). Results on the development set suggest utilizing instance-based methods with the FS6 feature set for this architecture (cf. "Full annotation" column in Table 2.13).

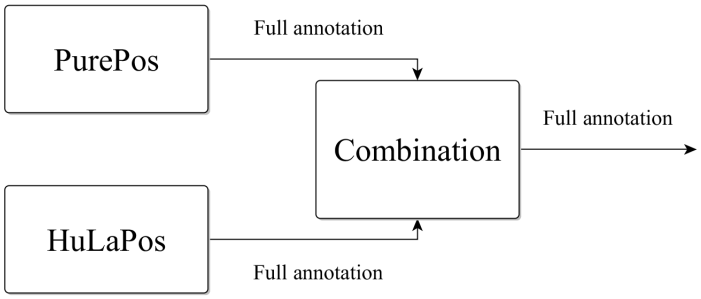


Figure 2.8 Combining the output of two morphological taggers

DOI:10.15774/PPKE.ITK.2015.007



#### 2.3.3.2 Combining PoS taggers only

Another plausible scheme is to combine only the PoS tagger modules of the tools (see Figure 2.9). However, in doing so, one has to deal with lemmatization as well. A straightforward solution for this to employ the lemmatizer of the better annotator tool (PurePos). Following this, the best tag selection model could be constructed (cf. Table 2.13) using instance based learning with the FS4 feature set.

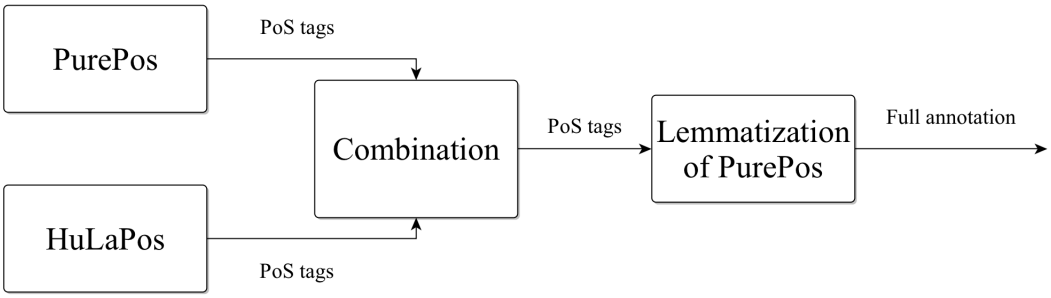


Figure 2.9 Combining the output of two PoS taggers and using also a lemmatizer

Although this algorithm allows us to create a better morpho-syntactic tagger compared to that of above, the gain in lemmatization remains much lower (6.81%). Consequently, the overall accuracy improvement measured in the development set (25.26%) is inferior.

#### 2.3.3.3 Multiple metalearners

Finally, the best results are produced using two level-1 learners: one of them chooses the better lemmatizer while the other selects the optimal PoS tagger (cf. Figure 2.10). In that way, this architecture can incorporate the best lemma and tag candidates (as in Table 2.13) yielding superior performance. However, a drawback of this configuration is that it may result in incompatible tag-lemma pairs[19]. To overcome this problem, this combination scheme is enhanced with the Humor morphological analyzer. This component is used to discover and fix incompatibilities. With this enhancement, we achieved 32.42% of improvement on the development set.

[19] A lemma and a tag for a word is incompatible if the MA can analyze the word, but no analysis contains both the lemma and the morpho-syntactic label.

DOI:10.15774/PPKE.ITK.2015.007



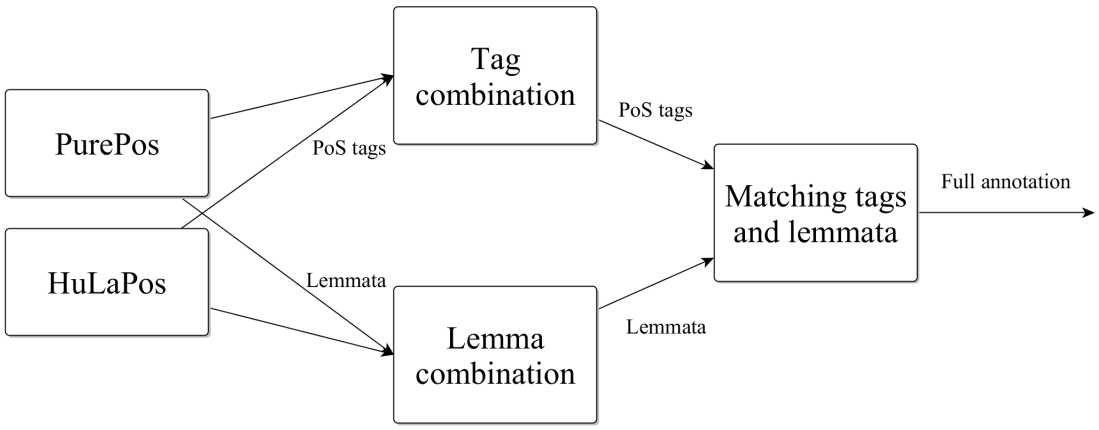


Figure 2.10 Combining the output of two PoS taggers and lemmatizers

### 2.3.4 Evaluation

Table 2.14 Relative error rate reduction on the test set compared to PurePos

| System | Tagging | Lemmatization | Full disamb. |
|---|---|---|---|
| *Oracle* | 48.60% | 59.42% | 51.53% |
| Disamb. combination | 23.23% | 23.55% | 26.86% |
| Tagger combination | 22.76% | 13.77% | 23.81% |
| Multiple metalearners | 25.07% | 29.89% | 28.90% |

All the presented combination schemes are evaluated on the unseen test set. Results show (cf. Table 2.14) that the hybrid architecture using a morphological analyzer achieved the best performance. While other schemes could also increase the performance of PurePos, it resulted in the highest accuracy (98.90%) fixing 28.90% of the baseline system's errors. This improvement score also shows that our system could capture more then half of the cases which can be fixed by a hypothetical oracle combinator. Further on, the proposed method also gives the highest error rate reduction in both PoS tagging and lemmatization. Concerning statistical significance, both the improvements of the presented schemes and their differences are significant at $p < 0.05$ (using Wilcoxon test of paired samples). These results show that our new combination

DOI:10.15774/PPKE.ITK.2015.007



architecture can be used in cases when very high disambiguation accuracy is crucial. Finally, we also confirmed that PurePos and HuLaPos can complement each other resulting in an improved morphological tagger.

DOI:10.15774/PPKE.ITK.2015.007

# 3

# An application of the tagger: estimating morpho-syntactic complexity

## 3.1 Motivation

Do linguists need to spend long hours counting morphemes for measuring morpho-syntactic complexity? Ever since the first studies, mean length of utterance (MLU) plays an important role in language development investigations. This metric has been widely used for measuring linguistic productivity of children for almost a hundred years. Utterance lengths are usually calculated in morphemes (MLUm) that is rather a time-consuming task. Even though CLAN toolkit [96, 97] can compute MLUm, this feature is only available just for a few languages not containing any agglutinative ones.

This chapter presents[1] an automatic method for estimating MLUm for Hungarian transcripts. We show that PurePos can be used effectively for aiding linguists in this scenario. Our approach adapts this tagging tool (cf. Section 2.2.2) yielding the first Hungarian tagger for spoken texts. Further on, we describe an MLUm estimation method, which is based on the adapted tagger, resulting in a high quality output.

[1] This study is a joint work with Kinga Jelencsik-Mátyus. Manual annotation of the data were performed by both of us, while the morpheme counting principles are her work. My contribution is the construction of the tagging chain, its adaptation and the automatization of the MLUm calculation.

DOI:10.15774/PPKE.ITK.2015.007



First, related studies are summarized, then we present the resources used for the research. Next, the adaptation steps of PurePos are described and the framework designed is introduced. Finally, we show that both the tagger and the estimator methods are accurate enough to replace the labor-intensive manual calculation.

## 3.2 Background

Tagging approaches of spoken languages mainly cover only mainstream ones (such as English, Italian or Spanish) while agglutinative ones are usually neglected. One of the pioneers in this field was Eeg-Oloffson [98] using manually annotated transcripts to train a statistical tagger (for English). In contrast, there are others employing and adapting statistics of written language corpora [99, 100, 101]. Besides, building domain-specific rules also lead to satisfactory taggers (e.g. [102]), while combination of such systems with stochastic tools [103] yields effective algorithms as well.

These previous studies imply that a proper morphological annotation system, which aims to process speech transcripts, must be able to handle the following types of difficulties:

1. existence of new morpho-syntactic tags which are missing from the tag-set of the training data,
2. occurrence of tokens with non-standard orthography in texts,
3. the number of words unknown to a statistical tagger are increased compared to written language corpora,
4. if probability estimates are derived from a written language training corpus, models of stochastic taggers can become non-representative (e.g. the distribution of PoS tags may significantly differ in written and spoken language).

Ever since the complexity of child language was measured, several methods have been developed. While manual counting prevailed for decades, automatic counting tools have been sought for in the past years.

DOI:10.15774/PPKE.ITK.2015.007



Several studies (e.g. [104]) showed that MLUm indicates language development for children, especially at very early stages. In contrast, mean length of utterance in words (MLUw) was shown to correlate highly [105, 106] with the latter in the case of analytical languages such as English or Irish. Therefore, some studies concur that MLUw is a reliable measure as opposed to MLUm, where researchers often need to make ad hoc decisions on what (not) to count (see [107]).

Crystal also points out [107] that computing length in morphemes is a good way to measure morphologically complex languages (see e.g. [108]). Hungarian is an agglutinative language, thus this measure can be considered to be a more reliable indicator of language development than MLUw (similarly to Turkish [109]). Moreover, previous studies investigating language development in Hungarian [110, 111] also employed MLUm as a metric.

In the case of corpora which follow the CHAT guidelines [97, 96], lengths of utterances (including morpheme counting) can be calculated [112] with the CLAN [96] toolkit. This system is widely used, since it has components performing the necessary preprocessing steps. One of its modules, MOR [96] is a morphological analyzer designed for spoken language corpora. A subsequent component is POST [113], doing the morphological disambiguation. Finally, a morpheme counter tool using their output is also available. In that way, MLUm is usually calculated in a number of languages applying these tools. However, they lack rules for Hungarian and many other morphologically complex languages, thus none of them can be used for analyzing such transcripts.

We are not aware of any research investigating the tagging of spoken Hungarian. Moreover, there is no study aiming to calculate MLUm for Hungarian transcripts automatically. Therefore, we introduce adaptation methods for a general-purpose tagger which is then utilized for counting morphemes resulting in accurate MLUm estimates.

DOI:10.15774/PPKE.ITK.2015.007



## 3.3 Resources

There is no Hungarian speech corpus morpho-syntactically annotated, therefore we use a contemporary one as a base of our research. Hungarian Kindergarten Language Corpus (HUKILC) [2] has been compiled predominantly for child language variation studies. It contains 62 interviews with 4.5–5.5 year-old kindergarten children from Budapest, recorded in the spring of 2012. The interviews are 20–30 minutes long consisting different types of story-telling tasks. Its transcription was carried out using the Child Language Data Exchange System (CHILDES) [97] following its guidelines. The corpus has about 39,000 utterances with 140,000 words.

In order to develop a proper tagger tool, a small part of the data has been manually annotated. As a first step, general tagging principles were established. We chose the morpho-syntactic labels and lemmata (detailed in [114]) of the Humor analyzer [29, 30] to represent morphological analyses. Next, an annotation manual was developed for human annotators to guide their work during the morphological disambiguation of the corpus. The whole process was carried out iteratively:

1. a portion of the sentences were morphologically disambiguated by the annotators independently,
2. discrepancies were discussed and resolved,
3. the annotation guide was updated accordingly.

In that way, 6 interviews with about 1,000 utterances were labeled manually by two experts (involving the author). Finally, the gold standard corpus was split randomly into two sets of equal sizes: a development and a test set (see Table 3.1).

Table 3.1 Number of tokens and utterances in the gold standard corpus

| | Utterances | Tokens |
|---|---|---|
| Development set | 509 | 3,340 |
| Test set | 449 | 2,740 |

DOI:10.15774/PPKE.ITK.2015.007



The tag-set of the corpus has been created to allow both the investigation of morpho-syntactic relations and the representation of phenomena typical to transcripts. First of all, a new label was introduced to mark filled pauses. Further on, the original annotation scheme of Humor distinguishes interjection and utterance words[2], but there are cases in speech when a word bears with both properties (such as *fúú* 'woow'). Therefore, a new label was created for annotating such tokens properly. Finally, the usage of diminutive is common in child transcripts, thus this property was indicated in labels and corresponding suffixes were omitted from lemmata.

## 3.4 Tagging children transcripts

The morphological tagging algorithm employed is a hybrid one. It is composed of a morphological analyzer, a stochastic tagger tool and several domain-specific disambiguation rules as well (cf. Figure 3.1). Since the tag-set of Humor was chosen to be used for the annotation, a plausible solution was to employ this analyzer. Further on, PurePos was utilized to disambiguate between the morphological annotation candidates. We used the Szeged Corpus [27] with Humor annotations to train the tagger.

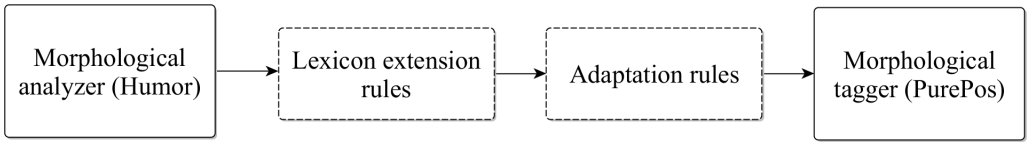


Figure 3.1 The architecture of the morphological tagging chain adapted for the HUKILC corpus

In order to apply a morphological analyzer prepared for written texts, its analyses had to be adjusted for the transcripts. Thus, adaptation rules – based on regular expressions and domain-specific word lists – were constructed using the development set. Their formulation could be done with high confidence, since most of the transcripts contained controlled conversation covering only a few topics.

[2] Annotation schemes for Hungarian distinguish utterances and interjection words. An utterance word forms a sentence or an utterance alone by interrupting or managing the communication. In contrast, interjections are either onomatopoeic or used to indicate emotions.

DOI:10.15774/PPKE.ITK.2015.007



As a first step, morphological analyses of about 40 words typical of spoken language were created manually. These tokens were mostly interjections not used in written language (such as *hűha* 'wow'), while some adverbs were regarded as utterance words in the corpus (e.g. *komolyan* 'seriously'). Furthermore, those tokens that are written as one word in transcripts but are spelled as two words in formal texts were also added to the lexicon. An example is *légyszíves* 'please' which is written formally as *légy szíves*. Finally, diminutive analyses were also provided where it was necessary. E.g. *kutyus* 'doggy' was also analysed as N.DIM with the lemma *kutya* 'dog' beside the old label N and the *kutyus* 'doggy' root. This process was carried out by investigating the lemmata produced by Humor: if the deletion of the derivational affix resulted in a root enumerated in a domain-specific list, a new diminutive analysis was created as well.

Concerning the disambiguation process, PurePos was extended with rules to adapt its knowledge to the target domain. First, the tagger was forced to assign diminutive analyses when it was possible. Secondly, rules were developed to assign interjection and utterance annotations for appropriate words. Finally, further enhancements were carried out by investigating the common mistakes of the tool on the development dataset.

A frequent error of the chain was the mistagging of *akkor* 'when' and *azért* 'in order to'. These words are pronouns and can be categorized as either adverbial, noun phrase level or demonstrative ones, and can also behave as pronomial adjectives. Generally, when *akkor* is followed by *amikor* 'when' (as in *Akkor érkezett meg, amikor mentem* 'He arrived, when I left') and when *azért* is followed by *mert* 'because' (as in the sentence *Azért eszik, mert éhes* 'He eats, because he is hungry') these pronouns are demonstrative ones. Furthermore, such co-occurances are more common in the transcripts than in the Szeged Corpus, since they are frequently used during reasoning or telling a story. As these long-term dependencies could not be learnt by the trigram tagger applied, rules were employed to tag these tokens correctly.

The next issue was the case of the word *utána* 'afterwards, then; after him/her/it'. It can either be used as an adverb of time (as in the sentence *Utána elindultunk* 'Then we left') and as a postpositional phrase meaning 'after him/her/it, following him/her/it'

DOI:10.15774/PPKE.ITK.2015.007



(as in *Elindultunk utána* 'We went after him'). The former usage is more frequent in spoken language: when this word is directly followed by conjunctions such as *meg* 'and' or *pedig* 'however', it is always an adverb. Therefore, *utána* was tagged as an adverb in the transcripts when it is followed by one of these trigger words.

The last rule introduced deals with *meg*, which may function as a verbal prefix or as a conjunction. Moreover, it is usually an expletive in spoken language. Therefore, the conjunctive label was assigned to the word when there was not any verb in its two token window.

## 3.5 Computing morpho-syntactic complexity

As a first step, general principles of counting morphemes were established. In a language with such a rich derivational system as Hungarian, it is often very complicated to identify the lemmata. This is even more difficult in our case, since no common methodology exists to determine the boundary of productivity in child language. This was based on studies of Brown [104], Retherford [115], Wéber [111] and Réger [110], with some necessary modifications. The basic principles were:

1. only meaningful words were analyzed, thus fillers (filled pauses such as *öööö* 'er'), punctuation marks and repetitions are not counted in the utterances;

2. phatic expressions (e.g. *igen, mhm* 'yes, uhm') serving to maintain communication and not conveying meaning were omitted;

3. inflectional suffixes and lemmata were each counted as one unit;

4. derivational morphemes (including diminutives) were not counted as separate ones,

5. reciprocal and indefinite pronouns (e.g. *minden#ki* 'everybody') and compound words (such as *kosár#labda* 'basketball') were counted as one morpheme.

DOI:10.15774/PPKE.ITK.2015.007



Following the guidelines of Brown [104], proper names (such as *Nagy Béla*, *Sári néni* 'Miss Sári') and lexicalized expressions (e.g. *Jó napot* 'Good morning'), which are frequent in speech, were also considered as one unit. Their identification was carried out employing rules. For this, the method relies on capitalized token sequences and a domain-specific list of words.

As for the automatization of rules, they were implemented using the morphological annotation of the corpus. First, each item on the list of fillers was eliminated. Afterwards, tagged words known to the MA were split into morphemes by the Humor analyzer regarding their computed morpho-syntactic annotation. If more than one analysis was available for a word in this phase, the least complex one was chosen, since analyses only differed in the number of derivative tags and compound markers (which we previously decided not to count) in the majority of the cases. As the labels of the annotation scheme were composed of morphemic properties, the estimation of unknown words could be based on their tags. Therefore, such calculations were carried out counting only the inflection markers in the guessed tags (as it is listed in principles above).

## 3.6 Evaluation

First of all, morpho-syntactic tagging performance of the system was investigated. Full analyses – containing both the lemmata and the tag – were compared to the gold standard data, not counting punctuation marks and hesitation fillers.

For measuring the individual advances of the enhancements presented, four different settings were evaluated on the test set. The first was a baseline using raw analyses of Humor disambiguated by PurePos. The second system (DIM) employed the extended vocabulary and handled the diminutive analyses as described in Section 3.4. The next one – marked with CONJ – utilized further rules aiming to tag *azért* and *amikor* correctly. Finally, the last system presented contains all the enhancements detailed above.

DOI:10.15774/PPKE.ITK.2015.007



Table 3.2 Evaluation of the improvements on the tagging chain (test set)

| Morph. tagger | Tagging accuracy | |
|---|---|---|
| | Token | Sentence |
| Baseline | 91.97% | 68.37% |
| + DIM | 94.92% | 79.96% |
| + CONJ | 95.53% | 81.74% |
| The full chain | 96.15% | 83.96% |

Measurements in Table 3.2 show that the baseline tool tagged erroneously 3 out of 10 sentences. On the contrary, each of the enhancements improved the overall performance significantly. For this, we used the Wilcoxon matched-pairs signed-rank test at $p<0.05$. Results indicate that the accuracy of the adapted chain is comparable with that of the tagging methods for written corpora [32]. Furthermore, the performance of the tool presented is similar or better compared to other results obtained by CLAN-based taggers: Parisse and Le Normand [113] report 5% error rate on annotating French transcripts, while the morphological disambiguator of Avaid et al. [116] results in 90% accuracy for Hebrew.

As for the MLU estimation task, two metrics were used for the evaluation. First, mean relative error was calculated (as in [42]), comparing the $a_i$ manual morpheme counts with $p_i$ predicted values for the $i$th utterances:

$$MRE = \sum_{i=1}^{n} \frac{|a_i - p_i|/a_i}{n} \tag{3.1}$$

This measurement shows the average relative deviation of the estimated morpheme counts from the one of human annotators.

DOI:10.15774/PPKE.ITK.2015.007



In addition, Pearson's correlation coefficient[3] (3.2) (cf. [42]) was employed as well:

$$\frac{S_{PA}}{\sqrt{S_P S_A}}, \text{ where} \tag{3.2}$$
$$S_{PA} = \frac{\sum_i (p_i - \overline{p})(a_i - \overline{a})}{n-1},$$
$$S_P = \frac{\sum_i (p_i - \overline{p})^2}{n-1} \text{ and } S_A = \frac{\sum_i (a_i - \overline{a})^2}{n-1}$$

This metric is used to measure the correspondence between the output of the processing chain and the counts of human annotators.

Table 3.3 Evaluation of the MLUm estimation algorithm using different morphological annotations

| Morphological annotation | MRE | Correlation |
|---|---|---|
| The output of the baseline tagger | 0.1325 | 0.9612 |
| The output of the adapted tagger | 0.0449 | 0.9901 |
| Gold standard annotations | 0.0279 | 0.9933 |

Since both metrics require a gold dataset, morpheme counts were manually calculated for 300 utterances of the test set. Table 3.3 presents the evaluation of the MLUm estimation algorithm on this manually checked corpus. First, we evaluated the output of the baseline tagger with our morpheme counter. Beside this, both the gold standard data and the output of the enhanced tagging tool were used as an input of the estimator. On the one hand, these results can be interpreted as an in vivo evaluation of the adapted tagger showing significant improvements over the baseline. On the other hand, it was found that the overall performance of the estimation methodology is outstandingly high. The high correlation of the automatic chain indicates that our method can properly measure the morpho-syntactic complexity of Hungarian spoken language in practice. Therefore, the time-consuming manual counting procedure can be replaced with the proposed method.

[3] The notation is the same above, except $\overline{x}$ is the average of $x_i$ values and $n$ denotes the total number of observation.

DOI:10.15774/PPKE.ITK.2015.007

# 4

# Methods for a less-resourced domain: preprocessing clinical Hungarian

## 4.1 Introduction

Hospitals produce a huge amount of clinical notes that have been solely used for archiving purposes and have generally been inaccessible to researchers. However, application of recent NLP technology can make accessible the hidden knowledge of archived records, thus boosting medical research. An example is the English cTAKES system [24], which can recognize important medical concepts in clinical free-text documents. Beside extracting diseases, symptoms and treatments, the tool is also able to identify various relations between them. This automatically extracted structured knowledge can be e.g. used to

- create family histories from medical records,
- find document discrepancies,
- spot similar cases and
- build question answering or semantic search systems.

While developing text processing tools for medicians is an emerging field in many developed countries, less-resourced languages lack such resources.

DOI:10.15774/PPKE.ITK.2015.007



To be able to extract information from medical texts, they must be preprocessed properly. Firstly, adequate text segmentation methods are required for finding token and sentence boundaries. Secondly, morphological tagging is an indispensable step for information extraction scenarios. Considering the case of Hungarian, there are only a few studies on processing medical records. Recently, Siklósi et al. [13, 117] have presented a system that is able to correct spelling errors in clinical notes. Their system uses a mixture of language models to generate correction candidates, however it focuses only on correctly segmented words. Beside error correction, an abbreviation resolution method was also presented by them [118], however, problems of text segmentation and morpho-syntactic tagging are still untouched. Furthermore, as far as we know, no study investigates such preprocessing tasks on Hungarian clinical texts.

Therefore, this chapter presents accurate preprocessing algorithms for noisy medical texts. Methods were developed and presented for only Hungarian, but they are designed in a way to perform well on other morphologically complex languages as well. Firstly, an effective method is introduced for detecting sentence and token boundaries. The presented system builds on well-known tokenization rules boosting them with the knowledge of a morphological analyzer and the output of an unsupervised filtering algorithm. Secondly, tagging experiments are presented yielding a viable morphological tagger for Hungarian electronic health records. The proposed tool builds on PurePos fixing its most common errors regarding the domain.

## 4.2 Segmenting texts of electronic health records

Error propagation in a text processing chain is usually a notable problem, therefore accurate segmentation methods are essential to parse texts properly. Moreover, notes written by doctors are extremely noisy containing errors which inhibit the application of existing tools.

DOI:10.15774/PPKE.ITK.2015.007



Even though tokenization and sentence segmentation methods perform well on general Hungarian, they have serious difficulties on clinical records. These originate in special properties of such texts involving

1. typing errors (i.e. mistyped tokens, nonexistent strings of falsely concatenated words) and

2. nonstandard usage of the language.

While errors of the first type can be corrected easily with e.g. a rule-based tool, others need advanced methods.

In this section, a hybrid approach to segmentation of noisy clinical records is presented. The method consists of two phases: first, tokens are partially segmented; then, sentence boundaries are identified. We start with detailing the background of our research and introducing resources used. Then, key elements of tokenization and SBD algorithms are described. Finally, our system is systematically evaluated on a gold standard corpus showing its high performance.

### 4.2.1 Previous approaches on text segmentation

Even though, numerous studies deal with English medical texts, only few attempts have been made (cf. [13, 117, 118]) for Hungarian. Further on, the task of detecting sentence and word boundaries in health records is often a neglected issue. Studies for Hungarian pay almost no attention on segmenting texts, while most of the approaches for English ignore this question. First, we review general tokenization and sentence boundary detection techniques first, then describe their application on the biomedical domain.

The task is often composed of several parts: normalization (when necessary), tokenization, and sentence boundary detection. Although, these are generally performed one after another, there are approaches (e.g. [119, 120]), where tokenization and SBD

DOI:10.15774/PPKE.ITK.2015.007



are treated as a unified tagging problem. Further on, handling of abbreviations is often involved in the segmentation process, since their identification helps to detect sentence and token boundaries.

As regards tokenization, it is generally treated as a simple engineering problem[1] cutting off punctuation marks from words. On the contrary, SBD is a rather researched topic. As Read et al. summarize [23], sentence segmentation approaches fall into three classes:

1. rule-based methods employing domain- or language-specific knowledge (such as abbreviations);

2. supervised machine learning approaches, which may not be robust amongst domains (being specialist on the training corpus); and

3. unsupervised learning methods extracting their knowledge from raw unannotated data.

As regards ML attempts, one of the first pioneers was Riley [121] who employed decision-tree learners to classify full stops. He utilized mainly lexical features (such as word length or case) to compute the probability of a word being sentence-initial or sentence-final. Next, Palmer et al. presented [122] the SATZ system, employing supervised learning algorithms. Since this tool can be easily adjusted through surface and syntactic features, it has been successfully applied to several European languages. Further on, the maximum entropy learning approach was used as well to the task by Reynar and Ratnaparkhi [123]. Their system classifies tokens containing '.', '?' or '!' characters utilizing contextual features and abbreviation lists. Recently, a similar approach has been presented by Gillick [124] for English, using support vector machines and resulting in state-of-the-art performance.

Beside machine learning approaches, rule-based methods are also commonly applied for these tasks. E.g. Mikheev introduced [125] a small set of rules for detecting sentence boundaries (SB) with a high accuracy. In another system presented of him

[1] In the case of alphabetic writing systems.

DOI:10.15774/PPKE.ITK.2015.007



[119], the latter method is integrated into a PoS tagging framework enabling the classification of punctuation marks. In doing so, they can be labeled as sentence boundaries, abbreviations or both. Moving on, Kiss and Strunk have introduced [40] an unsupervised method for sentence boundary detection in 2009. Their tool, Punkt uses scaled log-likelihood ratio for deciding whether a *(word, full stop)* pair is a collocation or not.

Although tokenization and SBD tasks are well established fields of natural language processing, there are only a few attempts aiming medical texts. These sentence segmentation attempts fall into two classes: some develop rule-based systems (e.g. [126]), while most of the studies employ supervised machine learning algorithms (such as [127, 128, 24, 129, 130]). Latter approaches usually train maximum entropy or CRF learners, thus large handcrafted training corpora are essential.

Training data used are either domain-specific or general. In practice, domain-specific knowledge yield better performance, however Tomanek et al. [131] argue on using only a general-purpose corpus. Their results indicate that the domain of the training corpus is not critical (at least for German).

As regards Hungarian, there are only two tools available. Huntoken [83] is an open source system based on Mikheev's system, while `magyarlanc` [32] has an adapted version of MorphAdorner's rule-based tokenizer [132] and sentence splitter. Both of them employ general-purpose methods utilizing language- and domain-specific rules and dictionaries.

This study introduces new methods for segmenting Hungarian clinical texts. For this, special properties of the target domain is investigated first by creating a manually segmented corpus. Then, a method is presented which combines high precision rules with unsupervised learning.

DOI:10.15774/PPKE.ITK.2015.007



### 4.2.2 Evaluation metrics

There is no metric commonly used to measure segmentation methods, therefore we review existing ones. On the one hand, researchers specializing in machine learning approaches prefer to calculate precision, recall and $F$-score. However, these measures are often used for computing the correctness of sentence boundaries only. On the other hand, studies on speech recognition prefer to compute NIST and Word Error Rate.

Recently, Read et al. have reviewed [23] the state-of-the-art of text segmentation proposing a unified metric to compare different approaches. Their method allows measuring sentence boundaries at any position labeling characters as sentence-finals or non sentence-finals. In doing so, simple accuracy measures the performance.

Our study builds on their results [23] adapting it to the full segmentation task of Hungarian clinical texts. In that way, we consider the corpus as a sequence of characters and empty strings and treat text segmentation as a single classification problem. Therefore, all the entities (either characters or empty string between them) can be labeled with one of the following tags:

$\langle$**T**$\rangle$ – if the entity is a token boundary,

$\langle$**S**$\rangle$ – if it is a sentence boundary,

$\langle$**None**$\rangle$ – otherwise.

This classification scheme enables us to calculate accuracy of the unified segmentation task. Moreover, it allows computing further common metrics as well.

Since it is important to measure each subsystem's correctness, precision and recall were calculated for both word tokenization and sentence boundary detection. Further on, word segmentation was evaluated with $F_1$, while $F_{0.5}$ was computed for the SBD task. (The latter calculation makes precision more important than recall.) We employed the latter metric, because an erroneously split sentence may cause information loss, while statements might still be extracted from longer multi-sentence text.

DOI:10.15774/PPKE.ITK.2015.007



### 4.2.3 Clinical texts used

A gold standard corpus of clinical texts was collected and manually corrected in order to develop and evaluate segmentation approaches. This process involved several steps involving normalization, as such texts are full with diverse mistakes. In doing so, we had to deal with the following types of errors[2]:

1. doubly converted characters, such as '>',
2. typewriter problems (e.g. '1' and '0' is written as 'l' and 'o'),
3. dates and date intervals being in various formats with or without necessary whitespaces (e.g. '2009.11.11', '06.01.08'),
4. missing whitespaces between tokens usually introduced various types of errors, such as:
   (a) measurements were erroneously attached to quantities (e.g. '0.12mg'),
   (b) lack of whitespace around punctuation marks (e.g. 'töröközegek.Fundus:ép.'),
5. various formulation of numerical expressions.

To investigate possible pitfalls, the gold standard data is split into two parts of equal sizes: a development and a test set containing 1,320 and 1,310 sentences respectively. The first part was used to identify typical problems and to develop the segmentation methods, while the second one was employed to evaluate the results.

As initial step, the distributions of abbreviations, punctuation marks and capitalization is investigated in these texts to reveal possible difficulties. Comparing our data with a corpus of general Hungarian (Szeged Corpus [27]) uncovers numerous discrepancies:

1. 2.68% of tokens found in the clinical corpus sample are abbreviations while the same ratio for general Hungarian is only 0.23%;

[2]Text normalization steps were carried out employing regular expressions.

DOI:10.15774/PPKE.ITK.2015.007



2. sentences taken from the Szeged Corpus almost always end in a sentence final punctuation mark (98.96%), while these are totally missing from clinical statements in 48.28% of the cases;

3. sentence-initial capitalization is a general rule in Hungarian (99.58% of the sentences are formulated properly in the Szeged Corpus), but its usage is not common in the case of clinicians (12.81% of the sentences start with a word that is not capitalized);

4. the amount of numerical data is notable in medical records (13.50% of sentences consist exclusively of measurement data and abbreviations), while text taken from the general domain rarely contains statements that are full of measurements.

### 4.2.4 Segmentation methods

Our system is built up from several components (cf. Figure 4.1). First, a symbolic method (referred as the baseline) marks word and sentence boundaries[3] seeking for full stops. Then, an unsupervised filtering method extends its output. Finally, rules employing capitalization yields further sentence boundaries.


Figure 4.1 The architecture of the proposed method

#### 4.2.4.1 Rule-based word tokenization and sentence segmentation

Our baseline method is composed of two parts. First, it tokenizes words (BWT) using regular expressions implemented in standard tokenizers. However, this algorithm does not try disambiguate all tokens containing periods, as it would need the proper recognition of domain-specific abbreviations as well.

[3]Rules and heuristics used are formulated investigating the development corpus.

DOI:10.15774/PPKE.ITK.2015.007



Further on, sentence segmentation (BSBD) is carried out minimizing information loss (as described in section 4.2.2). In that way, the method tries to avoid of making false-positive errors by splitting sentences only if there is a high confidence of success. We found that such cases are, when:

1. a period or exclamation mark directly follows another punctuation mark[4];
2. a line starts with a full date, and is followed by other words (The last white-space character before the date is marked as a sentence boundary (SB).);
3. a line begins with the name of an examination followed by a semicolon and a sequence of measurements.

Realization of these simple observations yield 100% precision and 73.38% recall on tokenization considering the development set. The corresponding values for detecting ends of sentences are 98.48% and 42.60% respectively. As less than half of the sentence boundaries are discovered, this method needs further improvements. In addition, a deeper analysis unfolded that the tokenization module has difficulties only with sentence final periods. We found that these sorts of errors are effects of the conservative tokenization algorithm, which left several words with punctuation mark attached ambiguous.

#### 4.2.4.2 Unsupervised sentence boundary classification

In order to enhance the baseline method we considered investigating two kinds of indicators that are usually employed in such scenarios:

**Periods:** when a punctuation mark (•) is attached to a word, a sentence boundary is found for sure only if the token is not an abbreviation.

**Capitalization:** if a word starts with a capital letter and it is neither part of a proper name nor of an acronym, it indicates the beginning of a sentence.

[4]Question marks are not considered as sentence-final punctuation marks, since they generally indicate a questionable finding in clinical texts.

DOI:10.15774/PPKE.ITK.2015.007



Considering our case:

1. clinicians introduce new abbreviations frequently which are not part of the standard, therefore a proper list cannot be collected easily, further on,

2. Latin words, abbreviations and subclauses are sometimes capitalized by mistake, thus they are neither reliable information sources.

In addition, numerous sentence boundaries lack both of these indicators (as shown in Section 4.2.3).

Even though these features do not function regularly, they can still be utilized. It is enough to find *evidence* for the separateness of a word and the subsequent full stop to classify a position as a sentence boundary. For this, we employed the idea of Kiss and Strunk [40] and adapted it for clinical texts.

The log-likelihood ratio method was first applied to identify collocations [39], however, Kiss and Strunk managed to adjust it for the SBD problem recently (cf. [40]). Their tool, called Punkt, considers abbreviations as collocations of words and periods, thus evaluating them using a modified log-likelihood ratio. In practice, this is formulated via a null hypothesis (4.1) and an alternative one (4.2).

$$H_0 : P(\bullet|w) = p = P(\bullet|\neg w) \tag{4.1}$$

$$H_A : P(\bullet|w) = p_1 \neq p_2 = P(\bullet|\neg w) \tag{4.2}$$

$$\log \lambda = -2 \log \frac{L(H_0)}{L(H_A)} \tag{4.3}$$

In these formulae, $H_0$ expresses the independence of a *(word, •)* pair, $H_1$ formulates that their co-occurrence is not just by chance, while $L$ denotes the likelihood function. The log-likelihood ratio of the null and alternative hypothesis is measured by the $\log \lambda$ score (4.3), which is found to be asymptotical to $\chi^2$ [39]. However, Kiss and Strunk

DOI:10.15774/PPKE.ITK.2015.007



recovered that this simple method performs poorly (in terms of precision) for identifying abbreviation. Therefore, they adapted the likelihood-ratio test by introducing several factors for scaling its result [40], and transforming it a heuristic ranking algorithm.

We improved their approach in numerous ways. First of all, the inverse score ($iscore = 1/log\lambda$) was used as a base, since it helps to find candidates co-occurring only by chance. Moving on, we introduced further scaling factors reviewing that of Punkt and adapting them to match the characteristics of the target domain.

First of all, the first factor of the Punkt system cannot be directly applied in our case. Counts and count ratios alone do not indicate properly alone whether a token and the period is related in a clinical record, since several sorts of abbreviations occur with relative low frequencies.

Next, lengths of words ($len$) was also used in Punkt to indicate abbreviations well. They could help in our case, since shorter tokens tend to be abbreviations, while longer ones do not. Therefore, we reformulated the original function to penalize short words and reward longer ones. Having a medical abbreviation list of almost 200 elements[5] we found that more than 90% of the abbreviations are shorter than three characters. This fact led us to formulate the scaling factor as in (4.4). In doing so, this enhancement can also decrease the score of a bad candidate, which distinguishes it from the original formula of Kiss and Strunk.

$$S_{length}(iscore) = iscore \cdot \exp(len/3 - 1) \tag{4.4}$$

Recently, Humor [29, 30, 31] has been extended with the content of a medical dictionary [16]. What is more, the analyzer is able indicate whether analyses refers to abbreviations. Therefore, its output is used to enhance the sentence segmentation algorithm.

[5]The list is gathered with an automatic algorithm on the development corpus using word shape properties and frequencies. The most frequent elements are manually verified and corrected.

DOI:10.15774/PPKE.ITK.2015.007



An indicator function was introduced (cf. Equation 4.5) to utilize its output deciding whether a word can be an abbreviation or not. Since, the morphological lexicon used is a well-established resource, our application could rely on it with high confidence. Therefore, the factor formulated (cf. Equation 4.6) uses larger weights compared to others. This method raises the score of a full word, decreases that of an abbreviation, while values of unknown words are left as they were.

$$indicator_{morph}(word) = \begin{cases} 1 & \text{if } word \text{ has an analysis of a known full word} \\ -1 & \text{if } word \text{ has an analysis of a known abbreviation} \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

$$S_{morph}(iscore) = iscore \cdot \exp\left(indicator_{morph} \cdot len^2\right) \tag{4.6}$$

Hyphens are generally not present in abbreviations but rather occurs in full words. Relying on this observation, *iscore* was adjusted (4.7) with a further indicator function: $indicator_{hyphen}$ outputs 1 only if the word contains a hyphen.

$$S_{hyphen}(iscore) = iscore \cdot \exp\left(indicator_{hyphen} \cdot len\right) \tag{4.7}$$

$$S = S_{length} \circ S_{morph} \circ S_{hyphen} \tag{4.8}$$

Scaled $\log \lambda$ (cf. $S(iscore)$ in Equation 4.8) is calculated for all *(word, •)* pairs not followed by any other punctuation mark. If this value is found to be higher than a threshold, the period is regarded as a sentence boundary and it is detached.[6] Otherwise, the joint token is treated as an abbreviation.

[6]Threshold value is empirically set to 1.5.

DOI:10.15774/PPKE.ITK.2015.007



To investigate the improvement of our method, it was pipelined with the BSBD module producing 77.14% recall and 97.10% precision on the development set. Accuracy values show significant improvements, however they also indicate that many sentence boundaries are still not found.

#### 4.2.4.3 Rules on capitalization

To further improve the method, capitalization properties of words were also utilized. We developed a rule-based component to decide whether a capitalized words can start a sentence or not. Good SB candidates of such tokens are the ones not following a non sentence terminating[7] punctuation, and are not part of a named entity. Therefore, sequences of capitalized words are considered to be named entities and omitted as a first step. Then, the rest of the candidates are processed by Humor. We employed a simple heuristic for detecting sentence boundaries: if a word does not have a proper noun analysis but is capitalized, it is marked as the beginning of a sentence. Our results on the development set show that this component also enhances the BSBD: it increases recall to 65.46% while keeps precision high (96.37%).

### 4.2.5 Evaluation

Table 4.1 Accuracy of the input text compared with the segmented ones

| | Accuracy |
|---|---|
| Raw corpus | 97.55% |
| BSBD | 99.11% |
| + LLR | 99.72% |
| + CAP | 99.26% |
| + LLR + CAP | 99.74% |

[7] Sentence terminating punctuation marks are the period and the exclamation mark for this task.

DOI:10.15774/PPKE.ITK.2015.007



Evaluation is presented for each components showing their accuracies (cf. Table 4.1). First, our improvements are compared to both the baseline module (BSBD) and the raw preprocessed corpus. The unsupervised SBD algorithm is marked with LLR[8], while the last component is indicated by CAP. Results show high accuracies for the overall segmentation task, furthermore the scores of the raw corpus is relatively high. This indicates that the metric applied is not well balanced.

Therefore, their improvements are also investigated calculating error rate reduction ratios (in Table 4.2). Comparison is carried out measuring enhancements over the baseline method (BSBD) showing that both of the components improves the segmentation method.

Table 4.2 Error rate reduction over the BSBD baseline method

| | Error rate reduction |
|---|---|
| LLR | 58.62% |
| CAP | 9.25% |
| LLR + CAP | 65.50% |

Considering sentence boundaries only, a more detailed analysis is got by computing precision, recall and $F_{0.5}$ values (in Table 4.3). Data shows that each component significantly increases the recall, while precision is just barely decreased. Finally, the combined hybrid algorithm[9] brings significant improvement over the well-established baseline.

Table 4.3 Precision, recall and F-score of the proposed sentence segmentation algorithms

| | Precision | Recall | $F_{0.5}$ |
|---|---|---|---|
| Baseline | 96.57% | 50.26% | 81.54% |
| + LLR | 95.19% | 78.19% | 91.22% |
| + CAP | 94.60% | 71.56% | 88.88% |
| + LLR + CAP | 93.28% | 86.73% | 91.89% |

[8]Referring to the term log-likelihood ratio.
[9]It is the composition of the BWT, BSBD, LLR and CAP components.

DOI:10.15774/PPKE.ITK.2015.007



While our approach focuses on the sentence identification task, we showed that it improves word tokenization as well. Table 4.4 presents measurements on word segmentation indicating that our enhancements resulted in a higher recall, while they did not decrease precision notably.

Table 4.4 Comparing the tokenization performance of the proposed tool with the baseline rule-based one

| | Precision | Recall | $F_1$ |
|---|---|---|---|
| Baseline | 99.74% | 74.94% | 85.58% |
| Hybrid system | 98.54% | 95.32% | 96.90% |

Besides, the proposed method was compared with freely available tools as well. There are only two applications for Hungarian text segmentation, which are `magyarlanc` and Huntoken. The latter system can be slightly adapted to a new domain by providing a set of abbreviations, thus two versions of it were evaluated. The first one employs a set of general Hungarian abbreviations (HTG), while the second one utilizes an extended dictionary[10] containing medical ones as well (HTM). Further on, Punkt [40] and the OpenNLP [82] toolkit[11] were also involved in our comparison. The latter tool is a general framework of maximum entropy methods, hence it could be applied to detect sentence boundaries as it is presented in [123].

[10] As described in section 4.2.4.2.

[11] The general-purpose Szeged Corpus was used as training data for the maximum entropy learning method.

DOI:10.15774/PPKE.ITK.2015.007



Table 4.5 Comparison of the proposed hybrid sentence segmentation method with other freely available tools

| | Precision | Recall | $F_{0.5}$ |
|---|---|---|---|
| `magyarlanc` | 72.59% | 77.68% | 73.55% |
| HTG | 44.73% | 49.23% | 45.56% |
| HTM | 43.19% | 42.09% | 42.97% |
| Punkt | 58.78% | 45.66% | 55.59% |
| OpenNLP | 52.10% | 96.30% | 57.37% |
| Hybrid system | 93.28% | 86.73% | 91.89% |

Results in Table 4.5 show that general segmentation methods fail on Hungarian clinical notes in contrast to our new algorithm. The hybrid approach presented bears with both high precision and recall, providing accurate sentence boundaries. While it was found that the maxent approach has decent recall as well, boundaries marked by it are false positives in almost half of the cases. Further on, rules of `magyarlanc` seem to be robust, but the overall low performance inhibits its application for clinical texts. Finally, other tools do provide not just low recalls, but their precision values are still around 50% limiting their applicability.

In sum, the presented segmentation method successfully deals with several sorts of imperfect sentence and word boundaries. It performs better in terms of precision and recall than competing ones, achieving 92% of $F_{0.5}$-score. Finally, our results indicates that the new hybrid algorithm is a proper tool for processing clinical Hungarian.

DOI:10.15774/PPKE.ITK.2015.007



## 4.3 Morphological tagging of clinical notes

Beside text segmentation, morphological tagging is also an indispensable task for information extraction scenarios. Even though tagging of general texts is well-known and considered to be solved, medical texts pose new challenges to researchers. In addition, English has been the main target of many studies investigating the biomedical domain up to the present time. Furthermore, there are just a few approaches for non-English data, neglecting agglutinative languages and particularly Hungarian.

This section investigates the tagging of clinical Hungarian by adapting existing methods. Our work is structured as follows. Related studies are described first, then a corpus of clinical notes is presented. Finally, domain adaptation enhancements are introduced which are then evaluated on the test corpus.

### 4.3.1 Background

In general, tagging of biomedical texts has an extensive literature, since numerous resources are accessible for English. On the contrary, much less manually annotated corpora of clinical texts are available. Further on, most of the work in this field was done for English, while only a few attempts were published for morphologically rich languages (e.g. [133, 134]).

First of all, a common approach for tagging biomedical text is to train supervised sequence-classifiers. However, a drawback of these methods is that they require manually annotated texts which are hard to create. Considering the types of training material, domain-specific corpora are used either alone [135, 24, 136] or in conjunction with a (sub)corpus of general English [137, 138, 139]. While utilizing texts only from the target domain yields acceptable performance [135, 24, 136], several experiments have shown that accuracy further increases with incorporating annotated sentences from the general domain as well [140, 137]. It is shown (e.g. [141]) that the more data is used from the reference domain, the higher accuracy can be achieved. However, Hahn and Wermter argue for training learners only on general corpora [142] (for German).

DOI:10.15774/PPKE.ITK.2015.007



Besides, there are studies on automatic selection of the training data (e. g. [143]). What is more, there are algorithms (such as [144]) learning from several domains parallelly thus delaying the model selection decision to the decoding process.

Next, utilization of domain-specific lexicons is another way of adapting taggers, as they can improve tagging performance significantly [137, 145]. Some studies extend existing PoS dictionaries [146], while others build new ones [136]. In brief, all such experiments yield significantly reduced error rates.

Concerning tagging algorithms, researchers tend to prefer already existing applications. One of the most popular system is the OpenNLP toolkit [82], which is e.g. the basis of the cTakes system [24]. Further on, TnT [35] is widely utilized (e.g. [142, 24]), and there are applications of Brill's method [78] as well (e.g. [141]). Besides, other HMM-based solutions were also shown to perform well [140, 137, 146, 142, 135, 134, 145] on biomedical texts.

Moving on, a number of experiments have revealed [138, 145, 136] that domain-specific OOV words are behind the reduced performance of taggers. Therefore, successful methods employ either guessing algorithms [140, 146, 134, 145, 136] or broad-coverage lexicons (as detailed above). Beyond supervised algorithms, other approaches were also shown to be effective: Miller et al. [139] used semi-supervised methods; Dwinedi and Sukhadeve built a tagger based only on rules [147]; while Ruch et al. proposed a hybrid system [145]. Further on, automatic domain adaptation methods (such as EasyAdapt [148], ClinAdapt [138] or reference distribution modelling [149]) also perform well. As a drawback, they need an appropriate amount of manually annotated data from the target domain limiting their applicability.

Our method builds on a baseline tagging chain composed of a trigram tagger (introduced in Section 2.2.2) and a broad coverage morphological analyzer. The latter tool employs a domain-adapted lexicon, while the tagger is adapted to the domain with further components.

DOI:10.15774/PPKE.ITK.2015.007



### 4.3.2 The clinical corpus

As there is no corpus of clinical records available manually annotated with morphological analyses, a new one was created. These texts contain about 600 sentences extracted from notes of 24 different clinics. First, textual parts of the records were identified (as described in [13]), then the paragraphs to be processed were selected randomly. After these, sentence boundary segmentation, tokenization and normalization was performed manually aided by methods of Section 4.2.4. Manual spelling correction was carried out relying on the system of Siklósi et al. [117]. Finally, morphological disambiguation was performed: the initial annotation was provided by PurePos, then its output was corrected manually.

As regards morphological annotation of texts, clinical notes have special properties differing from general Hungarian, which have been considered during their analysis. These texts contain numerous *x* tokens denoting multiplication, thus they are labeled as numerals. Latin words and abbreviations dominate sentences, which we decided to analyze regarding their meaning. For instance, *o.* denotes *szem* 'eye' thus it is tagged as a noun (N.NOM). Further on, medicine brand names are common as well, which were almost always found to be singular nouns. Finally, numerous sentences lack final punctuation marks that are not recovered in the test corpus.

The manually annotated corpus was split into two parts (cf. Table 4.6) for our experiments. The first one was employed for development purposes, while new methods were evaluated on the second part.

Table 4.6 Number of tokens ans sentences of the clinical corpus created

| | Sentences | Tokens |
|---|---|---|
| Development set | 240 | 2,230 |
| Test set | 333 | 3,155 |

These records are created in a special environment, thus they differ from general Hungarian in several aspects (cf. [8, 118, 13]):

DOI:10.15774/PPKE.ITK.2015.007



1. notes contain a lot of erroneously spelled words,
2. sentences generally lack punctuation marks and sentence initial capitalization,
3. measurements are frequent and have plenty of different (erroneous) forms,
4. a lot of (non-standard) abbreviations occur in such texts and
5. numerous medical terms are used originating from Latin.

### 4.3.3 The baseline setting and its most common errors

We built a baseline chain and analyzed its errors to improve the overall annotation quality. It uses the Humor analyzer, which produces *(morpho-syntactic tag, lemma)* pairs as analyses. (The output of the MA is extended with the new analysis of the *x* token to fit the corpus to be tagged.) Further on, analysis candidates are disambiguated by PurePos that is trained on the transcribed Szeged Corpus (as described in 2.2.3.1).

This baseline tagger produces 86.61% token accuracy[12] on the development set, which is remarkably lower than tagging results for general Hungarian using the same components (96–98% as in [6, 32]). Further on, sentence-based accuracy scores shows that less than the third (28.33%) of the sentences were tagged correctly. This fact indicates that the models of the baseline algorithm alone are weak for this task. Therefore, we investigated the most common errors of the chain.

Table 4.7 Distribution of the most frequent error types caused by the baseline algorithm (measured on the development set)

| Source of errors | Frequency | Ratio |
|---|---|---|
| Abbreviations and acronyms | 119 | 49.17% |
| Out-of-vocabulary words | 66 | 27.27% |
| Domain-specific PoS of word forms | 36 | 14.88% |

[12] Accuracy is calculated considering correct full analyses of tokens, not counting punctuation marks.

DOI:10.15774/PPKE.ITK.2015.007



Table 4.7 shows that the top error class is composed of mistagged abbreviations and acronyms. A reason for this is that most of the abbreviated tokens are previously not seen by the tagger. Therefore, their labels are produced by the tool's guesser module, which is not prepared for handling such tokens. What is more, these abbreviations usually refer to medical terms (and their inflected forms) originating from Latin, thus differing notably from standard ones.

Another class of mistakes was caused by out-of-vocabulary words. These are specific to the clinical domain and often originate from Latin. Although this observation is in accordance with the PoS tagging results for medical English, listing of such terms' analyses is not a satisfactory solution to the problem, since the number of inflected forms is significantly larger compared to English.

Finally, domain-specific usage of some words leads the tagger astray as well. An example is the class participles which are mislabeled as past tense verbs. E.g. *javasolt* 'suggested' and *felírt* 'written' are common words in the corpus, but have different PoS tag distributions in this domain. Further on, several erroneous tags are due to the lexical ambiguity being present in Hungarian (such as *szembe* which can refer to 'into an eye' or 'toward/against').

Based on the classification of errors above, domain-adaptation techniques were introduced enhancing the overall accuracy of the chain.

### 4.3.4 Domain adaptation experiments

#### 4.3.4.1 Utilizing an extended morphological lexicon

Supervised tagging algorithms commonly use augmented lexicons reducing the number of out-of-vocabulary words (see Section 4.3.1). In the case of Hungarian, this must be performed at the level of the morphological analyzer, since inflection is a momentous phenomenon. Extension of the lexicon was carried out by Attila Novák [15] adding 40,000 different lemmata to the analyzer. For this, he used a spelling dictionary of medical terms [150] and a freely available list of medicines [151]. By employing the enhanced lexicon, the ratio of OOV words was reduced to 26.19% (from 34.57%) that

DOI:10.15774/PPKE.ITK.2015.007



also improved the overall accuracy to 92.41% (on the development set). Further on, the medical dictionary [150] used contained numerous abbreviated tokens as well, thus the usage of the augmented analyzer also helped to decrease the number of mistagged abbreviations.

#### 4.3.4.2 Dealing with acronyms and abbreviations

Despite improvements above, numerous errors made by the enhanced tagger were still connected to abbreviations. Thus, we investigated the erroneous tags of abbreviated terms first, then methods were introduced for improving the performance of the disambiguation chain.

A detailed examination revealed that some erroneous tags were due to the over-generating nature of Humor. To fix such problems, we applied a simple filtering method. An analysis of a word with an attached full stop was considered to be a false candidate if the lemma candidate is not an abbreviation. Consequently, the overall accuracy was increased notably, reducing the number of errors on the development set by 9.20%.

Another typical error type was the mistagging of unknown acronyms. Since PurePos did not employ features dealing with such cases, these tokens were usually left to the suffix guesser resulting in incorrect annotation. In addition, our investigation shows that acronyms should be tagged as singular nouns in most of the cases. To annotate them properly, a pattern matching component was developed relying on surface features.

Finally, the rest of the errors were connected to those abbreviations which were both unknown to the analyzer and had not been seen previously. Therefore, the abbreviations labels was compared to that of the Szeged Corpus (see Table 4.8 below). While there are common properties between the two datasets (such as the ratio of adverbs), discrepancies are more significant. The most important difference is the proportions of adjectives: it is notably higher in the medical domain than in general Hungarian. Moreover, these values are more expressive if we consider that 10.85% of the tokens are abbreviated in the development set, while the same ratio is only 0.37% in the Szeged Corpus.

DOI:10.15774/PPKE.ITK.2015.007



Table 4.8 Morpho-syntactic tag frequencies of abbreviations on the development set

| Tag | Clinical texts | Szeged Corpus |
|---|---|---|
| N.NOM | 67.37% | 78.18% |
| A.NOM | 19.07% | 3.96% |
| CONJ | 1.27% | 0.50% |
| ADV | 10.17% | 11.86% |
| Other | 2.12% | 5.50% |

Since the nominal noun tag is the most frequent amongst abbreviations, a plausible method ("UnkN") was to assign the N.NOM label to unknown ones. Meanwhile, we kept the original word forms as lemmata. Although this approach is rather simple, it resulted in a surprisingly high (31.54%) error rate reduction (cf. Table 4.9).

Table 4.9 Accuracy scores of the abbreviation handling improvements on the development set

| ID | Method | Accuracy |
|---|---|---|
| 0 | Medical lexicon | 90.11% |
| 1 | 0 + Filtering | 91.02% |
| 2 | 1 + Acronyms | 91.41% |
| 3 | 2 + UnkN | 94.12% |
| 4 | 2 + UnkUni | 92.82% |
| 5 | 2 + UnkMLE | 94.01% |

Next, we tried to approximate the analyses of abbreviations with the distribution of tags observed in Table 4.8. First, we utilized ("UnkUni") a uniform distribution over their labels. The labels A.NOM, A.PRO, ADV, CONJ, N.NOM, V.3SG and V.PST_PTCL were used with equal probability as a sort of guessing algorithm.

Beside these, another reasonable method was to employ maximum likelihood estimation for calculating a priori probabilities of labels ("UnkMLE"). In that way, relative frequency estimates were computed for all the tags enlisted above.

DOI:10.15774/PPKE.ITK.2015.007



Comparing the performance of these enhancements (cf. Table 4.9), we found that this approach can also increase the overall performance, but the simple "UnkN" performs the best. This can be due to the fact that the data available could be insufficient for estimating probability distribution of labels properly.

#### 4.3.4.3 Choosing the proper training data

Since many studies showed (cf. Section 4.3.1) that the training data set significantly affects the result of a data-driven annotation chain, we investigated sub-corpora of the Szeged Corpus. Several properties (cf. Table 4.10) were examined[13] to find a decent domain to learn from for tagging clinical Hungarian.

Table 4.10 Comparing Szeged Corpus with clinical texts calculating average lengths of sentences, ratio of abbreviations and unknown words and perplexity regarding words and tags

| Corpus | Avg. sent. length | Abbrev. ratio | Unknown ratio | Perplexity | |
|---|---|---|---|---|---|
| | | | | Words | Tags |
| Szeged Corpus | 16.82 | 0.37% | 1.78% | 2318.02 | 22.56 |
| Fiction | 12.30 | 0.10% | 2.44% | 995.57 | 32.57 |
| Compositions | 13.22 | 0.14% | 2.29% | 1,335.90 | 30.78 |
| Computer | 20.75 | 0.14% | 2.34% | 854.11 | 22.89 |
| Newspaper | 21.05 | 0.20% | 2.10% | 1,284.89 | 22.08 |
| Law | 23.64 | 1.43% | 2.74% | 824.42 | 29.79 |
| Short business news | 23.28 | 0.91% | 2.50% | 859.33 | 27.88 |
| Development set | 9.29 | 10.85% | – | – | – |

First of all, an important attribute of texts is the length of sentences. Shorter sentences tend to have simpler grammatical structure, while longer ones are grammatically more complex. Further on, clinical texts have a vast amount of abbreviations, thus their ratio can also serve as a relevant metric. In addition, the

[13] Measurements regarding the development set were calculated manually where it was necessary.

DOI:10.15774/PPKE.ITK.2015.007



accuracy of a tagging system depends on the ratio of unknown words heavily, therefore their proportions were calculated. For this, we measured the ratio of OOV words on the development set.

Perplexity was also computed, since it can measure similarities of texts [152]. The calculation was carried out as follows: trigram models of word and tag sequences were trained on each corpus using Kneser-Ney smoothing, then all of them were evaluated on the development set[14].

Our examination shows that neither part of the Szeged Corpus contains as much abbreviated terms as clinical texts have. Likewise, sentences written by clinicians are significantly shorter than those of the Szeged Corpus. Neither the calculations above, nor the ratio of unknown words suggests using any of the subcorpora for training. However, the perplexity scores contradict: sentences from the law domain share the most phrases with clinical notes, while news texts have the most similar grammatical structures.

Table 4.11 Evaluation of the tagger on the development set trained with domain-specific subcorpora of the Szeged Corpus

| Corpus | Morph. disambiguation accuracy |
|---|---|
| Szeged Corpus | 94.73% |
| Fiction | 92.01% |
| Compositions | 91.97% |
| Computer | 92.73% |
| Newspaper | 93.29% |
| Law | 92.17% |
| Short business news | 92.69% |

Since similarity measurements were not in accordance with each other, all sub-corpora were tested as training data for tagging clinical texts. (These experiments were performed using the previously enhanced tagging chain.) The accuracy scores of

[14] We used the SRILM toolkit [153] for training models and measuring perplexity.

DOI:10.15774/PPKE.ITK.2015.007



taggers (cf. Table 4.11) on the development set show that training on a subcorpus cannot improve the performance. Therefore, we decided to use the whole Szeged Corpus to train our system.

### 4.3.5 Evaluation

The improved chain (cf. Table 4.12) was evaluated by investigating the part-of-speech tagging, lemmatization and the whole morphological annotation performance.

Table 4.12 Accuracy of the improved tagger on the test set

| ID | Method | PoS tagging | Lemmatization | Morph. disambig. |
|---|---|---|---|---|
| 0 | Baseline system | 90.57% | 93.54% | 88.09% |
| 1 | 0 + Lexicon extension | 93.89% | 96.24% | 92.41% |
| 2 | 1 + Handling abbreviations | 94.81% | 97.60% | 93.73% |

First of all, results show that the baseline method annotated almost 12% of the tokens erroneously, while our enhancements raised the ceiling of the full morphological tagging accuracy to 93.73%. Therefore, we managed to eliminate almost half (47.36%) of the errors. Next, accuracy scores also indicate that the error rate reduction is mainly due to the extended lexicon. However, the better handling of abbreviations also increased the performance significantly (Wilcoxon test of paired samples, $p < 0.05$). Therefore, our improvements yielded a system having satisfactory performance for morphologically parsing clinical texts.

This study revealed that abbreviations and out-of-vocabulary words cause the most of the errors for tagging Hungarian clinical texts. We introduced numerous enhancements dealing with them, although not all of them were successful. This could be due to the small amount of annotated data used inhibiting the better modeling of the domain.

DOI:10.15774/PPKE.ITK.2015.007

# 5

# Summary: new scientific results

## 5.1 New scientific results

### I Effective morphological tagging methods for morphologically rich languages

Full morphological tagging is a complex task composed of two parts. Beside identifying morpho-syntactic tags, lemmata of words must be computed as well. While the first task is a well-known problem of natural language processing, the latter one is often neglected. Results are summarized by describing the new lemmatization method first, followed by the full tagging systems.

**THESIS I.1. I developed a new lemmatization method for agglutinative languages. The presented algorithm is based on the output of a morphological analyzer. It can handle both known and unknown words effectively by incorporating diverse stochastic models. Results presented show that the new system has high accuracy on Hungarian texts.**

Publications: [18, 17, 10, 8]

DOI:10.15774/PPKE.ITK.2015.007



The proposed algorithm performs lemmatization in two steps. First, it uses a morphological analyzer and a guesser component to generate lemma candidates, then disambiguation is performed using stochastic models. The latter part is carried out calculating the score ($S$) of each lemma ($l$) for a given word ($w$) and tag ($t$) using the interpolation of two different models:

$$S(l|w,t) = P(l)^{\lambda_1} P(l,t|w)^{\lambda_2} \tag{5.1}$$

The system combines a simple unigram model with the output of a suffix-based guesser. To calculate the lambda parameters, guesses of models are evaluated on the training data, then the better model's score gets increased while that of the worse one is decreased.

Several experiments have been presented on the Szeged Corpus showing that the proposed method has superior accuracy for Hungarian compared to other available tools.

— • —

**THESIS I.2. I designed a hybrid morphological tagging system (PurePos[1]) for less-resourced and agglutinative languages. The method relies on stochastic methods incorporating the output of a morphological analyzer. Its lemmatization component utilizes algorithms presented in Thesis I.1. Furthermore, the tool is built up in a way to be able to incorporate domain-specific rules effectively. Experiments confirm its state-of-the-art accuracy for Hungarian and resource-scare scenarios.**

Publications: [18, 17, 10, 8]

The architecture of PurePos (cf. Figure 5.1) is built up to allow multiple models cooperating effectively. The disambiguation is carried out in multiple steps. The data flow starts from a MA providing word analyses as *(lemma, tag)* pairs. Next,

[1]The presented system is open source and is freely available at https://github.com/ppke-nlpg/purepos

DOI:10.15774/PPKE.ITK.2015.007



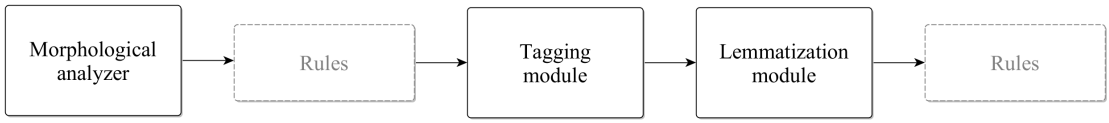


Figure 5.1 The architecture of the full morphological tagging tool

trigram-tagging methods (see [35, 36]) are employed for selecting morpho-syntactic labels of words. Finally, lemmatization is carried out employing the methods presented in Thesis I.1.

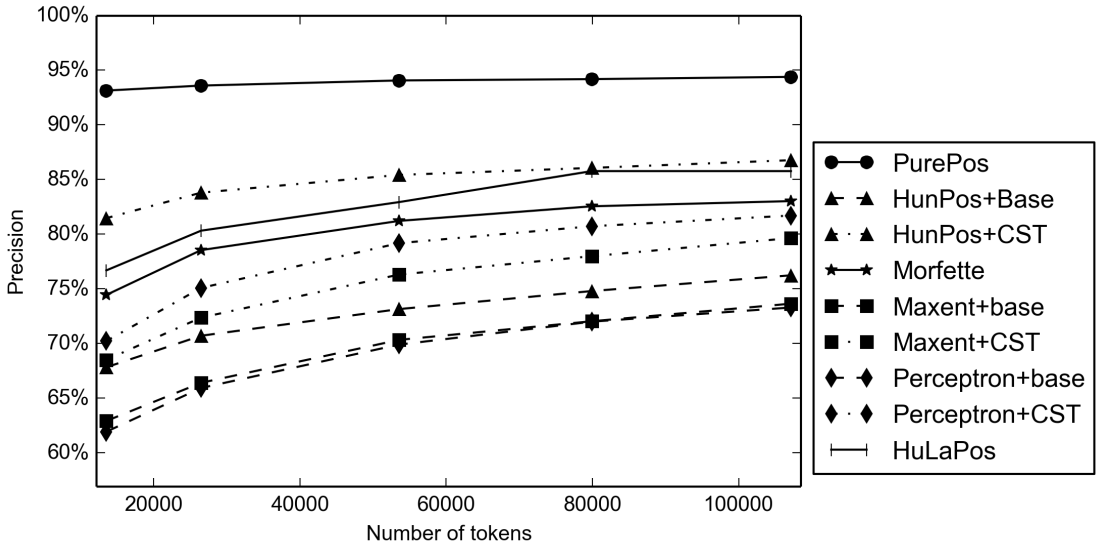


Figure 5.2 Learning curves of full morphological taggers on the Szeged Corpus (using Humor labels)

Several experiments were carried out measuring the performance of PurePos on the Szeged Corpus [27]. Results show that the new method yields very high (96.26%) full tagging accuracy on Hungarian. Moving on, I also compared existing tagging systems with the presented one on a less-resourced scenario. These experiments showed (cf. Figure 5.2) that PurePos can be successfully used even when the training dataset is limited. Finally, all the hybrid enhancements of PurePos ware evaluated one-by-one, showing that they can be used to fix several sorts of errors.

— • —

DOI:10.15774/PPKE.ITK.2015.007



Although, methods of Thesis group I.2 have high accuracy, it was shown that they can be improved further. Therefore, a combination technique is presented increasing the ceiling of morphological tagging tools' performance for agglutinative languages.

**THESIS I.3. I developed a methodology for combining morphological tagging systems effectively. The system presented selects the best lemma and tag candidates separately using two different combination methods. These components are trained with cross-validation using instance based learning. I showed that my method can significantly reduce the number of errors of existing annotation tools.**

Publications: [20, 9, 5]

First of all, discrepancy of tagging systems was analyzed. For this, I designed a new metric (Own Error Rate) which measures the differences of output of taggers. It turned out that the most typical mistakes of HuLaPos [7] and PurePos are different enough to be aggregated.

Following this, the most common combination techniques were investigated considering their applicability to full morphological tagging. Next, a new combination method was presented involving adapted feature sets for a morphologically rich language. It utilizes instance based learning [37] and trains classifiers with cross-validation, which can employ the whole training dataset for both the baseline tools and the level-one learners. The novelty of the presented method is its architecture (cf. Figure 5.3) which allows us to utilize different combiners for the lemmatization and PoS tagging subtasks.

DOI:10.15774/PPKE.ITK.2015.007



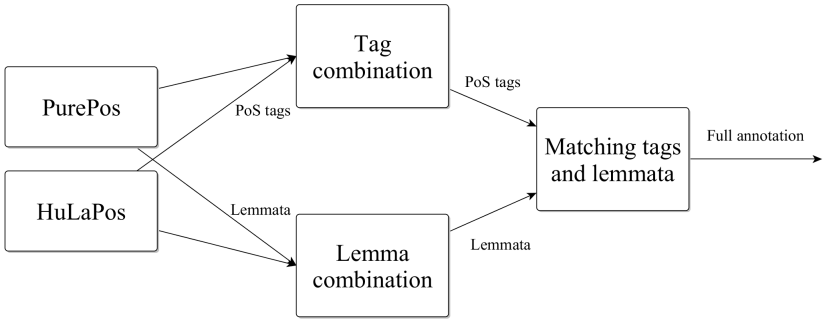


Figure 5.3 Combining the output of two PoS taggers and lemmatizers

Finally, evaluation experiments were presented indicating that the number errors of the best tagger can be decreased further. The new algorithm could reduce the number of errors of PurePos by 28.90%.

# II Measuring morpho-syntactic complexity using morphological annotation algorithms

Measuring morpho-syntactic complexity is usually carried out calculating mean length of utterances. This metric is often computed in words for analytical languages, while morphemes (MLUm) are used for morphologically complex ones. Although automatic methods and tools exist for e.g English, other less-resourced languages lack such systems. Therefore, MLUm could be only computed manually, which is a rather time-consuming task.

This thesis group presents[2] methods for processing speech transcripts effectively and estimating mean length of utterance in morphemes automatically.

**THESIS II.1. I developed a hybrid morphological tagging chain for Hungarian child-language transcripts. My method builds on top of the results presented in Thesis I.2 by adapting them to the domain. Evaluation**

[2]This research has been conducted together with Kinga Jelencsik-Mátyus. My contributions are the construction of the tagging chain, its adaptation and the automatization of the MLUm calculation.

DOI:10.15774/PPKE.ITK.2015.007



**shows that performance of the method is comparable with that of tagging methods for written corpora. Moreover, experiments indicate that the algorithm presented is accurate enough to be used in further applications.**

Publications: [2, 4]

The proposed method adapts the algorithms introduced in Thesis I.2 for spoken Hungarian. For this, the Humor morphological analyzer was augmented first with analyses of words typical to the domain. Next, the output of PurePos was adjusted utilizing domain-specific knowledge.

For this, a gold corpus of about 1,000 utterances from the HUKILC was created by the manual annotation of texts. Additionally, a new tagging scheme was designed representing the characteristics of spoken language properly.

The evaluation of the chain resulted in 96% token-level precision, which is comparable with that of taggers for corpora of written language. Therefore, my investigation showed that PurePos is an appropriate base for tagging corpora of transcribed spoken texts.

—•—

**THESIS II.2. I proposed a new algorithm for estimating morpho-syntactic complexity (calculating mean length of utterance in morphemes) in Hungarian child language transcripts. The method uses the morphological tagging chain of Thesis II.1 as a base. Evaluation of the system indicates that the methodology presented can properly replace the time-consuming manual computation of human annotators.**

Publications: [2, 4]

The estimation method analyzes morphological annotations of tokens. Words known by the analyzer are decomposed by Humor, while lengths of unknown words are guessed based on their PoS labels. This is followed by morpheme counting rules implementing linguistic guidelines, thus providing relevant estimates.

DOI:10.15774/PPKE.ITK.2015.007



As regards resources, a manually checked corpus was created for the experiments. Evaluation of the methods on this dataset shows that my results highly correlate (0.9901) with counts of human annotators. Further on, I showed that the mean relative error of the method is only 4.49%. Thus, the proposed algorithm can properly replace the labor-intensive human computation.

# III Effective preprocessing methods for a less-resourced noisy domain

More and more electronic health records are produced in hospitals containing valuable but hidden knowledge. Since doctors cannot spend enough time on writing their reports properly, notes often contain numerous errors. Because of such mistakes, processing of these texts cannot be carried out using general-purpose tools. Moreover, while several algorithms are becoming available for English, Hungarian and other morphologically rich languages are still neglected.

**THESIS III.1. I developed a new framework which segments noisy clinical records into words and sentences accurately. The method is built on top of well-known tokenization rules (e.g. [83]), however, it augments them with unsupervised heuristics. Evaluations showed that the algorithm can properly identify word and sentence boundaries in noisy clinical notes. Results also indicate that other systems available cannot handle such erroneous texts.**

Publications: [5, 14, 3]


Figure 5.4 The architecture of the proposed method

DOI:10.15774/PPKE.ITK.2015.007



The proposed method builds on pattern-matching algorithms taken from general-purpose tokenization tools. Even though these methods perform with high accuracy, their recall still stays low. Therefore, this study proposes a method (see Figure 5.4) which improves their performance using unsupervised heuristics and a domain-specific morphologic analyzer. First, the scaled $\log \lambda$ method [40] was adapted by introducing new scaling factors. Next, the Humor morphological analyzer was utilized to reveal further sentence boundaries.

The evaluation of the framework was carried out on a manually segmented corpus. Numerous metrics (such as precision, recall, F-score) were employed measuring the performance of the proposed tool. Moreover, existing Hungarian approaches were also compared with the proposed one.

Results show that other systems available can only produce low quality segmentation. Most of them yields F-scores less than 50% in sentence boundary identification. On the contrary, the method proposed can detect both token and sentence boundaries accurately, producing F-values over 90%.

—•—

**THESIS III.2. I showed that tagging methods of Thesis I.2 can be applied for annotating electronic health records satisfactorily. In doing so, PurePos was adjusted with stochastic and symbolic domain adaptation techniques. The quality of the annotation produced is comparable with that of general written tagger tools.**

Publications: [16, 1, 3]

First of all, an extended version of the Humor analyzer was used as a base of the tagging chain, since it was prepared[3] for electronic health records. Further on, the tagging chain was improved using a detailed error analysis of the baseline tagger.

[3]The lexicon extension was carried out by Attila Novák [15] .

DOI:10.15774/PPKE.ITK.2015.007



For this, a manually annotated corpus was created containing texts of clinical notes. Results on this dataset show that the improved system performs significantly better (93.73%) than the baseline system (88.09%). However, future work might target the segmentation and tagging tasks with a unified framework, since both systems have the most problems with abbreviated terms.

## 5.2 Applications

The methods presented here solve basic preprocessing tasks such as text segmentation and morphological tagging. Since these are essential components of any language processing chain, our results can be applied in numerous fields of natural language technology. In general, text mining solutions and information extraction tools utilize such algorithms. Since our methods aim morphologically rich and less-resourced languages (and especially Hungarian), they can be used to boost tasks involving such languages.

Concerning general tagging methods of Theses I.1 and I.2, they have been successfully applied in several Hungarian projects. Their applications involve the following studies:

1. Laki et al. [7] have developed an English to Hungarian morpheme-based statistical machine translation method using PurePos,

2. Novák et al. [12] have annotated Old and Middle Hungarian texts employing our methods,

3. Endrédy et al. [154] have proposed a noun phrase detection toolkit utilizing the morphological tagging tool presented,

4. Indig and Prószéky have applied [155] the proposed tagger tool for a batch spelling-correction tool and

5. Prószéky et al. [156] have built their psycho-linguistically motivated parser on top of PurePos.

DOI:10.15774/PPKE.ITK.2015.007



Next, Thesis group II presents methods and resources for analyzing transcripts of spoken language which can serve NLP applications of the domain. Besides, methods of Thesis II.2 estimate morpho-syntactic complexity of children language, thus can replace the labor-intense manual work. Furthermore, Jelencsik-Mátyus utilizes [157] these algorithms in her research investigating the language development of Hungarian kindergarten children.

Finally, the last (III) Thesis group details methods for processing noisy texts effectively. Algorithms of Thesis III.1 segment clinical texts accurately, providing proper output for information extraction applications. Furthermore, lessons learned from our tagging methods could help the development of accurate text mining tools in the target domain. Besides, an ongoing project [158, 159, 3] on processing Hungarian electronic health records benefits from the proposed methods.

DOI:10.15774/PPKE.ITK.2015.007

# Bibliography

## The author's journal publications

[1] **György Orosz**, Attila Novák, and Gábor Prószéky. Lessons learned from tagging clinical Hungarian. *International Journal of Computational Linguistics and Applications*, 5(1):159–176, 2014. ISSN: 0976-0962.

[2] Kinga Mátyus and **György Orosz**. MONYEK: morfológiailag egyértelműsített óvodai nyelvi korpusz. *Beszédkutatás – 2014*:237–245, 2014. ISSN: 1218-8727.

[3] Borbála Siklósi, Attila Novák, **György Orosz**, and Gábor Prószéky. Processing noisy texts in Hungarian: a showcase from the clinical domain. *Jedlik Laboratories Reports*, II(3):5–62, 2014. Péter Szolgay, editor. ISSN: 2064-3942.

## The author's book section publications

[4] **György Orosz** and Kinga Mátyus. An MLU Estimation Method for Hungarian Transcripts. English. In Petr Sojka, Aleš Horák, Ivan Kopeček, and Karel Pala, editors, *Text, Speech and Dialogue*. Volume 8655, in Lecture Notes in Computer Science, pages 173–180. Springer International Publishing. ISBN: 978-3-319-10815-5.

DOI:10.15774/PPKE.ITK.2015.007



[5] **György Orosz**, Attila Novák, and Gábor Prószéky. Hybrid text segmentation for Hungarian clinical records. In Félix Castro, Alexander Gelbukh, and Miguel González, editors, *Advances in Artificial Intelligence and Its Applications*. Volume 8265, in Lecture Notes in Computer Science, pages 306–317. Springer, Berlin Heidelberg, 2013. ISBN: 978-3-642-45114-0.

[6] **György Orosz**, László János Laki, Attila Novák, and Borbála Siklósi. Improved Hungarian Morphological Disambiguation with Tagger Combination. In Ivan Habernal and Václav Matousek, editors, *Text, Speech, and Dialogue*. Volume 8082, in Lecture Notes in Computer Science, pages 280–287. Springer, Berlin, Heidelberg, 2013. ISBN: 978-3-642-40584-6.

[7] László János Laki, **György Orosz**, and Attila Novák. HuLaPos 2.0 – Decoding Morphology. In Félix Castro, Alexander Gelbukh, and Miguel González, editors, *Advances in Artificial Intelligence and Its Applications*. Volume 8265, in Lecture Notes in Computer Science, pages 294–305. Springer Berlin Heidelberg, 2013. ISBN: 978-3-642-45113-3.

# The author's international conference publications

[8] **György Orosz** and Attila Novák. PurePos 2.0: a hybrid tool for morphological disambiguation. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2013)*. INCOMA Ltd. Shoumen, Hissar, Bulgaria, 2013, pages 539–545.

[9] **György Orosz**, László János Laki, Attila Novák, and Borbála Siklósi. Combining Language Independent Part-of-Speech Tagging Tools. In *2nd Symposium on Languages, Applications and Technologies*. José Paulo Leal, Ricardo Rocha, and Alberto Simões, editors. In OpenAccess Series in Informatics (OASIcs). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Porto, 2013, pages 249–257.

DOI:10.15774/PPKE.ITK.2015.007



[10] **György Orosz** and Attila Novák. PurePos – an open source morphological disambiguator. In *Proceedings of the 9th International Workshop on Natural Language Processing and Cognitive Science*. Bernadette Sharp and Michael Zock, editors. Wroclaw, 2012, pages 53–63.

[11] László Laki and **György Orosz**. An Efficient Language Independent Toolkit for Complete Morphological Disambiguation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland, 2014, pages 26–31.

[12] Attila Novák, **György Orosz**, and Nóra Wenszky. Morphological annotation of Old and Middle Hungarian corpora. In *Proceedings of the ACL 2013 workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*. Piroska Lendvai and Kalliopi Zervanou, editors. Association for Computational Linguistics, Sofia, Bulgaria, 2013, pages 43–48.

[13] Borbála Siklósi, **György Orosz**, Attila Novák, and Gábor Prószéky. Automatic structuring and correction suggestion system for Hungarian clinical records. In *8th SaLTMiL Workshop on Creation and use of basic lexical resources for less-resourced languages*. Guy De Pauw, Gilles-Maurice de Schryver, Mike L. Forcada, Francis M. Tyers, and Peter Waiganjo Wagacha, editors. Istanbul, 2012, pages 29–34.

# The author's other conference publications

[14] **György Orosz** and Gábor Prószéky. Hol a határ? Mondatok, szavak, klinikák. In *X. Magyar Számítógépes Nyelvészeti Konferencia*. Attila Tanács, Viktor Varga, and Veronika Vincze, editors. Szegedi Tudományegyetem, Szeged, 2014, pages 177–187.

DOI:10.15774/PPKE.ITK.2015.007



[15] **György Orosz** and Attila Novák. PurePos 2.0: egy hibrid morfológiai egyértelműsítő rendszer. In *X. Magyar Számítógépes Nyelvészeti Konferencia*. Attila Tanács, Viktor Varga, and Veronika Vincze, editors. Szegedi Tudományegyetem, Szeged, 2014, pages 373–377.

[16] **György Orosz**, Attila Novák, and Gábor Prószéky. Magyar nyelvű klinikai rekordok morfológiai egyértelműsítése. In *IX. Magyar Számítógépes Nyelvészeti Konferencia*. Attila Tanács and Veronika Vincze, editors. Szegedi Tudományegyetem, Szeged, 2013, pages 159–169.

[17] **György Orosz**. PurePos: hatékony morfológiai egyértelműsítő. In *VI. Alkalmazott Nyelvészeti Doktoranduszkonferencia*. Tamás Váradi, editor. Budapest, 2012, pages 134–139.

[18] **György Orosz**, Attila Novák, and Balázs Indig. Javában taggelünk. In *VIII. Magyar Számítógépes Nyelvészeti Konferencia*. Attila Tanács and Veronika Vincze, editors. Szeged, 2011, pages 336–340.

[19] László János Laki and **György Orosz**. HuLaPos2 – Fordítsunk morfológiát. In *X. Magyar Számítógépes Nyelvészeti Konferencia*. Attila Tanács, Viktor Varga, and Veronika Vincze, editors. Szegedi Tudományegyetem, Szeged, 2014, pages 41–49.

[20] László János Laki and **György Orosz**. Morfológiai egyértelműsítés nyelvfüggetlen annotáló módszerek kombinálásával. In *IX. Magyar Számítógépes Nyelvészeti Konferencia*. Attila Tanács and Veronika Vincze, editors. Szegedi Tudományegyetem, Szeged, 2013, pages 331–337.

[21] Borbála Siklósi, **György Orosz**, and Attila Novák. Magyar nyelvű klinikai dokumentumok előfeldolgozása. In *VIII. Magyar Számítógépes Nyelvészeti Konferencia*. Attila Tanács and Veronika Vincze, editors. Szeged, 2011, pages 143–154.

DOI:10.15774/PPKE.ITK.2015.007



# Other references

[22] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999. ISBN: 978-0-262-13360-9.

[23] Jonathon Read, Rebecca Dridan, Stephan Oepen, and Lars Jørgen Solberg. Sentence Boundary Detection: A Long Solved Problem? In *24th International Conference on Computational Linguistics (Coling 2012)*. India, 2012.

[24] Guergana K. Savova, James J. Masanz, Philip V. Ogren, Jiaping Zheng, Sunghwan Sohn, Karin Kipper Schuler, and Christopher G. Chute. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513, 2010.

[25] Tomaž Erjavec. MULTEXT-East: Morphosyntactic Resources for Central and Eastern European Languages. *Language resources and evaluation*, 46(1):131–142, March 2012. ISSN: 1574-020X.

[26] Bart Jongejan and Hercules Dalianis. Automatic training of lemmatization rules that handle morphological changes in pre- , in- and suffixes alike. In *Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, 2009, pages 145–153. ISBN: 978-1-932432-45-9.

[27] Dóra Csendes, János Csirik, and Tibor Gyimóthy. The Szeged Corpus: A POS tagged and syntactically annotated Hungarian natural language corpus. In *Proceedings of the 5th International Workshop on Linguistically Interpreted Corpora LINC 2004 at The 20th International Conference on Computational Linguistics COLING 2004*, 2004, pages 19–23.

DOI:10.15774/PPKE.ITK.2015.007



[28] Attila Novák and Nóra Wenszky. Ó- és középmagyar szóalaktani elemző. In *IX. Magyar Számítógépes Nyelvészeti Konferencia*. Attila Tanács and Veronika Vincze, editors. Szegedi Tudományegyetem, Informatikai Tanszékcsoport, Szeged, 2013, pages 170–181.

[29] Gábor Prószéky and Balázs Kis. A Unification-based Approach to Morpho-syntactic Parsing of Agglutinative and Other (Highly) Inflectional Languages. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*. Robert Dale and Kenneth Ward Church, editors. ACL, College Park, Maryland, 1999, pages 261–268.

[30] Attila Novák. Milyen a jó humor? In *Magyar Számítógépes Nyelvészeti Konferencia 2003*. Szeged, 2003, pages 138–145.

[31] Gábor Prószéky and Attila Novák. Computational Morphologies for Small Uralic Languages. In *Inquiries into Words, Constraints and Contexts.* Stanford, California, 2005, pages 150–157.

[32] János Zsibrita, Veronika Vincze, and Richárd Farkas. magyarlanc: A Toolkit for Morphological and Dependency Parsing of Hungarian. In *Proceedings of Recent Advances in Natural Language Provessing 2013*. Association for Computational Linguistics. Hissar, Bulgaria, 2013, pages 763–771.

[33] Lawrence Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[34] Christer Samuelsson. Morphological tagging based entirely on Bayesian inference. In *9th Nordic Conference on Computational Linguistics NODALIDA-93*. Stockholm University, Stockholm, Sweden, 1993.

DOI:10.15774/PPKE.ITK.2015.007



[35] Thorsten Brants. TnT – A Statistical Part-of-Speech Tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*. Universität des Saarlandes, Computational Linguistics. Association for Computational Linguistics, 2000, pages 224–231.

[36] Péter Halácsy, András Kornai, and Csaba Oravecz. HunPos: an open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*. Association for Computational Linguistics, Prague, Czech Republic, 2007, pages 209–212.

[37] David W. Aha, Dennis Kibler, and Marc K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.

[38] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009. ISSN: 19310145.

[39] Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational linguistics*, 19(1):61–74, 1993.

[40] Tibor Kiss and Jan Strunk. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525, 2006.

[41] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: open source scientific tools for Python. [Online; accessed 2014-11-26]. 2001–. URL: http://www.scipy.org/.

[42] Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd edition, 2011, page 629. ISBN: 978-0-12-374856-0.

[43] Walter S. Stolz, Percy H. Tannenbaum, and Frederick V. Carstensen. A Stochastic approach to the grammatical coding of English. *Communications of the ACM*, 8(6):399–405, 1965.

DOI:10.15774/PPKE.ITK.2015.007



[44] Sheldon Klein and Robert F. Simmons. A Computational Approach to Grammatical Coding of English Words. *Journal of the ACM*, 10(3):334–347, July 1963. ISSN: 0004-5411.

[45] Djame Seddah, Sandra Koebler, and Reut Tsarfaty, editors. *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*. Association for Computational Linguistics, Los Angeles, CA, USA, 2010.

[46] Djamé Seddah, Reut Tsarfaty, and Jennifer Foster, editors. *Proceedings of the Second Workshop on Statistical Parsing of Morphologically Rich Languages*. Association for Computational Linguistics, Dublin, Ireland, 2011.

[47] Marianna Apidianaki, Ido Dagan, Jennifer Foster, Yuval Marton, Djamé Seddah, and Reut Tsarfaty, editors. *Proceedings of the ACL 2012 Joint Workshop on Statistical Parsing and Semantic Processing of Morphologically Rich Languages*. Association for Computational Linguistics, Jeju, Republic of Korea, 2012.

[48] Yoav Goldberg, Yuval Marton, Ines Rehbein, and Yannick Versley, editors. *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*. Association for Computational Linguistics, Seattle, Washington, USA, 2013.

[49] Csaba Oravecz and Péter Dienes. Efficient Stochastic Part-of-Speech Tagging for Hungarian. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*. European Language Resources Association (ELRA), Las Palmas, Canary Islands - Spain, 2002.

[50] Martin F. Porter. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3):130–137, 1980.

[51] Joel Plisson, Nada Lavrac, and Dunja Mladenić. A rule based approach to word lemmatization. In *SiKDD 2004 at Multiconference IS-2004*. Ljubljana, Slovenia, 2004, pages 83–86.

DOI:10.15774/PPKE.ITK.2015.007



[52] Matjaž Juršič, Igor Mozetič, and Nada Lavrač. Learning Ripple Down Rules for Efficient Lemmatization. In *Proceedings of the 10th International Multiconference Information Society*. Dunja Mladenić and Marko Grobelnik, editors. IJS, Ljubljana, Slovenia, October 2007, pages 206–209.

[53] Haşim Sak, Tunga Güngör, and Murat Saraçlar. Morphological Disambiguation of Turkish Text with Perceptron Algorithm. English. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*. Volume 4394, in Lecture Notes in Computer Science, pages 107–118. Springer Berlin Heidelberg. ISBN: 978-3-540-70938-1.

[54] Georgiana Dinu Grzegorz Chrupała and Josef van Genabith. Learning Morphology with Morfette. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, and Daniel Tapias, editors. European Language Resources Association (ELRA), Marrakech, Morocco, 2008, pages 2362–2367. ISBN: 2-9517408-4-0.

[55] Jan Hajič and Barbora Hladká. Tagging inflective languages: Prediction of morphological categories for a rich, structured tagset. In *Proceedings of the 17th international conference on computational linguistics*, 1998, pages 483–490.

[56] Dan Tufis and Oliver Mason. Tagging Romanian texts: a case study for Qtag, a language independent probabilistic tagger. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC)*, 1998, pages 589–596.

[57] Mükka Silfverberg and Krister Lindén. Combining Statistical Models for POS Tagging using Finite-State Calculus. In *Proceedings of the 18th nordic conference of computational linguistics nodalida 2011*, 2011, pages 183–190.

DOI:10.15774/PPKE.ITK.2015.007



[58] Tomaz Erjavec and Saso Dzeroski. Machine Learning of Morphosyntactic Structure: Lemmatizing Unknown Slovene Words. *Applied Artificial Intelligence*, 18(1):17–41, 2004.

[59] Željko Agić, Nikola Ljubešić, and Danijela Merkler. Lemmatization and Morphosyntactic Tagging of Croatian and Serbian. In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*. Association for Computational Linguistics, Sofia, Bulgaria, August 2013, pages 48–57.

[60] Dilek Z. Hakkani-Tür, Kemal Oflazer, and Gökhan Tür. Statistical Morphological Disambiguation for Agglutinative Languages. *Computers and the humanities*, 36(4):381–410, November 2002. ISSN: 1572-8412.

[61] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*. In ACL '07. Association for Computational Linguistics, Prague, Czech Republic, 2007, pages 177–180.

[62] Maciej Piasecki and Adam Wardyński. Multiclassifier Approach to Tagging of Polish. *Proceedings of the international multiconference on computer science and information technology*, 1:169–178, 2006.

[63] Maciej Piasecki. Polish tagger TaKIPI: Rule based construction and optimisation. *Task quarterly*, 11(1-2):151–167, 2007.

[64] Szymon Acedanski. A Morphosyntactic Brill Tagger for Inflectional Languages. In *IceTAL*. Hrafn Loftsson, Eiríkur Rögnvaldsson, and Sigrún Helgadóttir, editors. Volume 6233. In Lecture Notes in Computer Science. Springer, 2010, pages 3–14. ISBN: 978-3-642-14769-2.

DOI:10.15774/PPKE.ITK.2015.007



[65] Adam Radziszewski. A Tiered CRF Tagger for Polish. English. In Robert Bembenik, Lukasz Skonieczny, Henryk Rybinski, Marzena Kryszkiewicz, and Marek Niezgodka, editors, *Intelligent Tools for Building a Scientific Information Platform*. Volume 467, in Studies in Computational Intelligence, pages 215–230. Springer Berlin Heidelberg. ISBN: 978-3-642-35646-9.

[66] Adam Radziszewski and Tomasz Śniatowski. A Memory-Based Tagger for Polish. In *Proceedings of the LTC 2011*, 2011.

[67] Drahomíra Spoustová, Jan Hajič, Jan Votrubec, Pavel Krbec, and Pavel Květoň. he best of two worlds: Cooperation of statistical and rule-based taggers for Czech. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing: Information Extraction and Enabling Technologies*. In ACL '07. Association for Computational Linguistics, 2007, pages 67–74.

[68] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 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*. Marti Hearst and Mari Ostendorf, editors. Association for Computational Linguistics, Edmonton, Canada, 2003, pages 173–180.

[69] Georgi Georgiev, Valentin Zhikov, Petya Osenova, Kiril Simov, and Preslav Nakov. Feature-rich Part-of-speech Tagging for Morphologically Complex Languages: Application to Bulgarian. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. In EACL '12. Association for Computational Linguistics, Avignon, France, 2012, pages 492–502. ISBN: 978-1-937284-19-0.

[70] Hrafn Loftsson. Tagging Icelandic text using a linguistic and a statistical tagger. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*. Association for Computational Linguistics, 2007, pages 105–108.

DOI:10.15774/PPKE.ITK.2015.007



[71] Turhan Daybelge and Ilyas Cicekli. A Rule-Based Morphological Disambiguator for Turkish". In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2007), Borovets*, 2007, pages 145–149.

[72] Joseph Le Roux, Benoit Sagot, and Djamé Seddah. Statistical Parsing of Spanish and Data Driven Lemmatization. In *Proceedings of the ACL 2012 Joint Workshop on Statistical Parsing and Semantic Processing of Morphologically Rich Languages*. Association for Computational Linguistics, Jeju, Republic of Korea, 2012, pages 55–61.

[73] Kepa Bengoetxea, Koldo Gojenola, and Arantza Casillas. Testing the effect of morphological disambiguation in dependency parsing of basque. In *Proceedings of the Second Workshop on Statistical Parsing of Morphologically Rich Languages*. Association for Computational Linguistics, 2011, pages 28–33.

[74] Wolfgang Maier, Sandra Kübler, Daniel Dakota, and Daniel Whyatt. Parsing German: How Much Morphology Do We Need? In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*. Dublin City University, Dublin, Ireland, 2014, pages 1–14.

[75] Bernd Bohnet, Joakim Nivre, Igor Boguslavsky, Richárd Farkas, Filip Ginter, and Jan Hajič. Joint morphological and syntactic analysis for richly inflected languages. *Transactions of the Association for Computational Linguistics*, 1:415–428, 2013.

[76] Thomas Müller, Helmut Schmid, and Hinrich Schütze. Efficient Higher-Order CRFs for Morphological Tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington, USA, 2013, pages 322–332.

[77] Beáta Megyesi. Brill's rule-based part of speech tagger for Hungarian. Master's thesis. Stockholm: Stockholm University, 1998, page 136.

DOI:10.15774/PPKE.ITK.2015.007



[78] Eric Brill. A simple rule-based part of speech tagger. *Proceedings of the Third Conference on Applied Natural Language Processing*, 28(4):152–155, 1992. ISSN: 00992399.

[79] Tamás Horváth, Zoltán Alexin, Tibor Gyimóthy, and Stefan Wrobel. Application of Different Learning Methods to Hungarian Part-of-Speech Tagging. English. In Sašo Džeroski and Peter Flach, editors, *Inductive Logic Programming*. Volume 1634, in Lecture Notes in Computer Science, pages 128–139. Springer Berlin Heidelberg. ISBN: 978-3-540-66109-2.

[80] Péter Halácsy, András. Kornai, Csaba Oravecz, Viktor Trón, and Dániel Varga. Using a morphological analyzer in high precision POS tagging of Hungarian. In *Proceedings of the Fifth conference on International Language Resources and Evaluation*, 2006, pages 2245–2248.

[81] András Kuba, András Hócza, and János Csirik. POS Tagging of Hungarian with Combined Statistical and Rule-Based Methods. English. In Petr Sojka, Ivan Kopeček, and Karel Pala, editors, *Text, Speech and Dialogue*. Volume 3206, in Lecture Notes in Computer Science, pages 113–120. Springer Berlin Heidelberg, 2004. ISBN: 978-3-540-23049-6.

[82] Jason Baldridge, Thomas Morton, and Gann Bierner. The OpenNLP maximum entropy package. 2002. URL: http://maxent.sourceforge.net.

[83] Péter Halácsy, András Kornai, László Németh, András Rung, István Szakadát, and Viktor Trón. Creating open language resources for Hungarian. In *In Proceedings of the 4th International Conference on Language Resources and Evaluation*, 2004, pages 1201–1204.

[84] Fernando Enríquez and Fermín L. Cruz and F. Javier Ortega and Carlos G. Vallejo and José A. Troyano. A comparative study of classifier combination applied to NLP tasks. *Information Fusion*, 14(3):255–267, 2013. ISSN: 1566-2535.

DOI:10.15774/PPKE.ITK.2015.007



[85] Lars Kai Hansen and Peter Salamon. Neural Network Ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990. ISSN: 0162-8828.

[86] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004, page 350. ISBN: 0-471-21078-1.

[87] Lei Xu, Adam Krzyzak, and Ching Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(3):418–435, 1992. ISSN: 0018-9472.

[88] Eric Brill and Jun Wu. Classifier combination for improved lexical disambiguation. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*. Volume 1. Association for Computational Linguistics, Stroudsburg, PA, USA, 1998, pages 10–18.

[89] Hans Van Halteren, Jakub Zavrel, and Walter Daelemans. Improving Accuracy in Word Class Tagging through the Combination of Machine Learning Systems. *Computational Linguistics*, 27(2):199–229, 2001. ISSN: 08912017.

[90] Anders Søgaard. *Ensemble-based POS tagging of Italian*. In. *Poster and Workshop Proceedings of the 11th Conference of the Italian Association for Artificial Intelligence*. Italian Association for Artificial Intelligence, 2009. ISBN: 978-88-903581-1-1.

[91] Tomasz Śniatowski and Maciej Piasecki. Combining Polish Morphosyntactic Taggers. In Pascal Bouvry, Mieczysław A. Kłopotek, Franck Leprévost, Małgorzata Marciniak, Agnieszka Mykowiecka, and Henryk Rybiński, editors, *Security and intelligent information systems*. Volume 7053, in Lecture Notes in Computer Science, pages 359–369. Springer Berlin Heidelberg, 2012. ISBN: 978-3-642-25260-0.

DOI:10.15774/PPKE.ITK.2015.007



[92] Maytham Alabbas and Allan Ramsay. Improved POS-Tagging for Arabic by Combining Diverse Taggers. In Lazaros Iliadis, Ilias Maglogiannis, and Harris Papadopoulos, editors, *Artificial Intelligence Applications and Innovations*. Volume 381, in IFIP Advances in Information and Communication Technology, pages 107–116. Springer Berlin Heidelberg, 2012. ISBN: 978-3-642-33408-5.

[93] Lars Borin. Something Borrowed, Something Blue: Rule-Based Combination of POS Taggers. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC 2000)*, 2000, pages 21–26.

[94] Jan Hajič, Pavel Krbec, Květoň, Karel Oliva, and Vladimír Petkevič. Serial combination of rules and statistics: A case study in Czech tagging. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2001, pages 268–275.

[95] George H. John and Pat Langley. Estimating Continuous Distributions in Bayesian Classifiers. In *Eleventh Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, San Mateo, 1995, pages 338–345.

[96] Brian MacWhinney. The CHILDES project: Tools for analyzing talk: Volume I: Transcription format and programs, Volume II: The database. *Computational Linguistics*, 26(4):657, 2000.

[97] Brian MacWhinney. CHAT manual – The Childes Project. Online; accessed 1-December-2014. 2014. URL: http://childes.talkbank.org/manuals/chat.pdf.

[98] Mats Eeg-Olofsson. Word-Class Tagging: Some Computational Tools. PhD thesis. Göteborg, Sweden: University of Göteborg: Department of Computational Linguistics, 1991, page 99.

[99] Amália Mendes, Raquel Amaro, and M. Fernanda Bacelar do Nascimento. Morphological tagging of a spoken Portuguese corpus using available resources. In *Language technology for Portuguese: shallow processing tools and resources*. António Branco, Amália Mendes, and Ricardo Ribeiro, editors. Lisboa: Colibri, 2004, pages 47–62.

DOI:10.15774/PPKE.ITK.2015.007



[100] Joakim Nivre, Leif Grönqvist, Malin Gustafsson, Torbjörn Lager, and Sylvana Sofkova. Tagging spoken language using written language statistics. In *Proceedings of the 16th Conference on Computational linguistics – Volume 2*. Association for Computational Linguistics, 1996, pages 1078–1081.

[101] Alessandro Panunzi, Eugenio Picchi, and Massimo Moneglia. Using PiTagger for Lemmatization and PoS Tagging of a Spontaneous Speech Corpus: C-ORAL-ROM Italian. In *4th Language Resource and Evaluation Conference (LREC)*, 2004, pages 563–566.

[102] Antonio Moreno and José M. Guirao. Tagging a spontaneous speech corpus of Spanish. In *Proceedings of Recent Advances in Natural Language Processing (RANPL) 2003*. Borovets, Bulgaria, 2003, pages 292–296.

[103] Eckhard Bick, Heliana Mello, Alessandro Panunzi, and Tommaso Raso. The annotation of the C-ORAL-BRASIL oral through the implementation of the Palavras Parser. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. ELRA, Istanbul, Turkey, 2012.

[104] Roger Brown. *A first language: The early stages*. Harvard University Press, Oxford, 1973, page 437.

[105] Tina Hickey. Mean length of utterance and the acquisition of Irish. *Journal of Child Language*, 18(3):553–569, 1991.

[106] Matthew D. Parker and Kent Brorson. A comparative study between mean length of utterance in morphemes (MLUm) and mean length of utterance in words (MLUw). *First Language*, 25(3):365–376, 2005.

[107] David Crystal. Review of R. Brown 'A first language'. *Journal of Child Language*, 11:289–307, 1974.

[108] Melissa Bowerman. *Early syntactic development: A cross-linguistic study with special reference to Finnish*. Cambridge University Press, Cambridge, 1973, page 307. ISBN: 0521200199.

DOI:10.15774/PPKE.ITK.2015.007



[109] Ayşe Pınar Saygın. A Computational Analysis of Interaction Patterns in the Acquisition of Turkish. *Research on language and computation*, 8(4):239–253, 2010.

[110] Zita Réger. Mothers' speech in different social groups in Hungary. *Children's language*, 7:197–222, 1990.

[111] Katalin Wéber. "Rejtelmes kétféleség" – A kétféle igeragozás elkülönülés a magyar nyelvben. Hungarian. PhD thesis. Pécs: University of Pécs, 2011, page 218.

[112] Kenji Sagae, Eric Davis, Alon Lavie, Brian Macwhinney, and Shuly Wintner. Morphosyntactic annotation of CHILDES transcripts. *Journal of child language*, 37(3):705–729, 2010.

[113] Christophe Parisse and Marie-Thérese Le Normand. Automatic disambiguation of morphosyntax in spoken language corpora. *Behavior Research Methods, Instruments and Computers*, 32(3):468–481, 2000.

[114] Attila Novák. Description of the format and semantics of the output of the present version of the Hungarian Humor analyzer. Technical report. Budapest, Hungary: MorphoLogic Ltd., 2003.

[115] Kristine S. Retherford. *Guide to analysis of language transcripts*. Thinking Publications, 3rd edition, 1993, page 283. ISBN: 1888222417.

[116] Aviad Albert, Brian MacWhinney, Bracha Nir, and Shuly Wintner. The Hebrew CHILDES corpus: transcription and morphological analysis. *Language resources and evaluation*, 47(4):973–1005, 2013.

[117] Borbála Siklósi, Attila Novák, and Gábor Prószéky. Context-Aware Correction of Spelling Errors in Hungarian Medical Documents. In Adrian-Horia Dediu, Carlos Martín-Vide, Ruslan Mitkov, and Bianca Truthe, editors, *Statistical Language and Speech Processing*. Volume 7978, in Lecture Notes in Computer Science, pages 248–259. Springer Berlin Heidelberg, 2013. ISBN: 978-3-642-39592-5.

DOI:10.15774/PPKE.ITK.2015.007



[118] Borbála Siklósi and Attila Novák. *Detection and Expansion of Abbreviations in Hungarian Clinical Notes*. In. *MICAI 2013: 12th Mexican International Conference on Artificial Intelligence*. Volume 8265. In Lecture Notes in Artificial Intelligence. Springer-Verlag, Heidelberg, 2013, pages 318–328.

[119] Andrei Mikheev. Tagging sentence boundaries. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*. Association for Computational Linguistics, 2000, pages 264–271.

[120] Conghui Zhu, Jie Tang, Hang Li, Hwee Tou Ng, and Tiejun Zhao. A unified tagging approach to text normalization. In *The 45st Annual Meeting of the Association for Computational Linguistics*, 2007, pages 688–695.

[121] Michael D. Riley. Some applications of tree-based modelling to speech and language. In *Proceedings of the Workshop on Speech and Natural Language*. Association for Computational Linguistics, 1989, pages 339–352.

[122] David D Palmer and Marti A Hearst. Adaptive multilingual sentence boundary disambiguation. *Computational linguistics*, 23(2):241–267, 1997.

[123] Jeffrey C. Reynar and Adwait Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the fifth Conference on Applied Natural Language Processing*. Association for Computational Linguistics, 1997, pages 16–19.

[124] Dan Gillick. Sentence boundary detection and the problem with the US. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*. Association for Computational Linguistics, 2009, pages 241–244.

[125] Andrei Mikheev. Periods, capitalized words, etc. *Computational Linguistics*, 28(3):289–318, 2002.

DOI:10.15774/PPKE.ITK.2015.007



[126] Hua Xu, Shane P. Stenner, Son Doan, Kevin B. Johnson, Lemuel R. Waitman, and Joshua C. Denny. MedEx: a medication information extraction system for clinical narratives. *Journal of the American Medical Informatics Association*, 17(1):19–24, 2010.

[127] Emilia Apostolova, David S. Channin, Dina Demner-Fushman, J. Furst, S. Lytinen, and D. Raicu. Automatic segmentation of clinical texts. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*. IEEE, 2009, pages 5905–5908.

[128] Paul S. Cho, Ricky K. Taira, and Hooshang Kangarloo. Text boundary detection of medical reports. In *Proceedings of the AMIA Symposium*. American Medical Informatics Association, 2002, page 998.

[129] Ricky K. Taira, Stephen G. Soderland, and Rex M. Jakobovits. Automatic Structuring of Radiology Free-Text Reports. *Radiographics*, 21(1):237–245, 2001.

[130] Katrin Tomanek, Joachim Wermter, and Udo Hahn. Sentence and token splitting based on conditional random fields. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, 2007, pages 49–57.

[131] Katrin Tomanek, Joachim Wermter, and Udo Hahn. A reappraisal of sentence and token splitting for life sciences documents. *Studies in Health Technology and Informatics*, 129(Pt 1):524–528, 2006.

[132] Amit Kumar. MONK Project: Architecture Overview. Technical report. Northwestern University, 2009, page 3.

[133] Michel Oleynik, Percy Nohama, Pindaro Secco Cancian, and Stefan Schulz. Performance analysis of a POS tagger applied to discharge summaries in Portuguese. *Studies in health technology and informatics*, 160(2):959–963, 2009.

DOI:10.15774/PPKE.ITK.2015.007



[134] Thomas Brox Røst, Ola Huseth, Øystein Nytrø, and Anders Grimsmo. Lessons from Developing an Annotated Corpus of Patient Histories. *Journal of computing science and engineering*, 2(2):162–179, 2008.

[135] Serguei V. S. Pakhomov, Anni Coden, and Christopher G. Chute. Developing a corpus of clinical notes manually annotated for part-of-speech. *International Journal of Medical Informatics*, 75(6):418–429, September 13, 2009.

[136] Lawrence H. Smith, Thomas C. Rindflesch, and W. John Wilbur. The importance of the lexicon in tagging biological text. *Natural language engineering*, 12(4):335–351, 2006. ISSN: 1351-3249.

[137] Anni Coden, Sergey V. Pakhomov, Rie Kubota Ando, Patrick H. Duffy, and Christopher G. Chute. Domain-specific language models and lexicons for tagging. *Journal of biomedical informatics*, 38(6):422–430, July 5, 2006.

[138] Jeffrey P. Ferraro, Hal III Daumé, Scott L. DuVall, Wendy W. Chapman, Henk Harkema, and Peter J. Haug. Improving performance of natural language processing part-of-speech tagging on clinical narratives through domain adaptation. *Journal of the American Medical Informatics Association*:931–939, 2013. ISSN: 1067-5027.

[139] John Miller, Manabu Torii, and K Vijay-Shanker. Building Domain-Specific Taggers without Annotated (Domain) Data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2007, pages 1103–1111.

[140] Neil Barrett and Jens Weber-Jahnke. A Token Centric Part-of-Speech Tagger for Biomedical Text. In Mor Peleg, Nada Lavrač, and Carlo Combi, editors, *Artificial Intelligence in Medicine*. Volume 6747, in Lecture Notes in Computer Science, pages 317–326. Springer Berlin Heidelberg, 2011. ISBN: 978-3-642-22217-7.

DOI:10.15774/PPKE.ITK.2015.007



[141] John Pestian, Lukasz Itert, and Wlodzislaw Duch. Development of a Pediatric Text-Corpus for Part-of-Speech Tagging. In *Intelligent Information Processing and Web Mining*. Mieczyslaw A. Klopotek, Slawomir T. Wierzchon, and Krzysztof Trojanowski, editors. In Advances in Soft Computing. Springer, July 9, 2004, pages 219–226. ISBN: 3-540-21331-7.

[142] Udo Hahn and Joachim Wermter. Tagging Medical Documents with High Accuracy. In *PRICAI 2004: Trends in Artificial Intelligence*. Chengqi Zhang, Hans W. Guesgen, and Wai-Kiang Yeap, editors. Volume 3157. In Lecture Notes in Computer Science. Springer, 2004, pages 852–861. ISBN: 3-540-22817-9.

[143] Kaihong Liu, Wendy Chapman, Rebecca Hwa, and Rebecca S. Crowley. Heuristic sample selection to minimize reference standard training set for a part-of-speech tagger. *Journal of the American Medical Informatics Association*, 14(5):641–650, 2007.

[144] Jinho D. Choi and Martha Palmer. Fast and Robust Part-of-Speech Tagging Using Dynamic Model Selection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. Association for Computational Linguistics. The Association for Computer Linguistics, 2012, pages 363–367. ISBN: 978-1-937284-25-1.

[145] Patrick Ruch, Robert Baud, Pierrette Bouillon, and Gilbert Robert. Minimal commitment and full lexical disambiguation: Balancing rules and hidden markov models. In *Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2000, pages 111–114.

[146] Guy Divita, Allen C. Browne, and Russell Loane. dTagger: a POS tagger. In *AMIA Annual Symposium Proceedings*. American Medical Informatics Association, 2006, pages 200–203.

DOI:10.15774/PPKE.ITK.2015.007



[147] Sanjay Kumar Dwivedi and Pramod P. Sukhadeve. Rule-based Part-of-speech Tagger for Homoeopathy Clinical Realm. *IJCSI International Journal of Computer Science*, 8(4):350–354, July 2011. ISSN: 1694-0814.

[148] Hal Daumé III. Frustratingly Easy Domain Adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics, Prague, Czech Republic, June 2007, pages 256–263.

[149] Yuka Tateisi, Yoshimasa Tsuruoka, and Jun-ichi Tsujii. Subdomain adaptation of a POS tagger with a small corpus. In *Proceedings of the Workshop on Linking Natural Language Processing and Biology: Towards Deeper Biological Literature Analysis*. Association for Computational Linguistics, 2006, pages 136–137.

[150] Pál Fábián and Péter Magasi. *Orvosi helyesírási szótár*. Péter Magasi, editor. Akadémiai Kiadó, Budapest, 1992, page 591. ISBN: 9789630562980.

[151] Országos Gyógyszerészeti Intézet Főigazgatóság. Forgalomba hozatali engedéllyel rendelkező allopátiás és homeopátiás készítmények. http://www.ogyi.hu/generalt_listak/tk_lista.csv. Online; accessed 20-December-2012.

[152] Adam Kilgarriff and Tony Rose. Measures for corpus similarity and homogeneity. In *Proceedings of the Third Conference on Empirical Methods for Natural Language Processing*. Nancy Ide and Atro Voutilainen, editors. Association for Computational Linguistics. Granada, Spain, 1998, pages 46–52.

[153] Andreas Stolcke. SRILM – an extensible language modeling toolkit. In *Proceedings International Conference on Spoken Language Processing*. John H. L. Hansen and Bryan L. Pellom, editors. ISCA, November 2002, pages 257–286.

DOI:10.15774/PPKE.ITK.2015.007



[154] István Endrédy. Corpus driven research: ideas and attempts. In *PhD Proceedings Annual Issues of the Doctoral School - 2014*. Faculty of Information Technology and Bionics, Pázmány Péter Catholic University., Budapest, Hungary, 2014, 137–140.

[155] Balázs Indig and Gábor Prószéky. Ismeretlen szavak helyes kezelése kötegelt helyesírás-ellenőrző programmal. In *X. Magyar Számítógépes Nyelvészeti Konferencia*. Attila Tanács, Viktor Varga, and Veronika Vincze, editors. Szegedi Tudományegyetem, Szeged, 2013, pages 310–317.

[156] Gábor Prószéky, Balázs Indig, Márton Miháltz, and Bálint Sass. Egy pszicholingvisztikai indíttatású számítógépes nyelvfeldolgozási modell felé. In *X. Magyar Számítógépes Nyelvészeti Konferencia*. Attila Tanács, Viktor Varga, and Veronika Vincze, editors. Szegedi Tudományegyetem, Szeged, 2014, pages 79–87.

[157] Kinga Jelencsik-Mátyus. A szociolingvisztikai stílus: Stratégiák a gyermek-felnőtt diskurzusban. PhD thesis. Szeged, Hungary: University of Szeged, 2015, page 192.

[158] Borbála Siklósi and Attila Novák. Identifying and clustering relevant terms in clinical records using unsupervised methods. In Laurent Besacier, Adrian-Horia Dediu, and Carlos Martín-Vide, editors, *Statistical language and speech processing*. Volume 8791, in Lecture Notes in Computer Science, pages 233–243. Springer International Publishing, 2014. ISBN: 978-3-319-11396-8.

[159] Borbála Siklósi and Attila Novák. A magyar beteg. In *X. Magyar Számítógépes Nyelvészeti Konferencia*. Attila Tanács, Viktor Varga, and Veronika Vincze, editors. Szegedi Tudományegyetem, Szeged, 2014, pages 188–198.