MorphoTrees of Arabic and Their Annotation in the TrEd Environment

Otakar Smrž[Σ], **Petr Pajas**[Π]

smrz@ckl.mff.cuni.cz, pajas@ufal.mff.cuni.cz

[Σ]Center for Computational Linguistics, Faculty of Mathematics and Physics, Charles University in Prague
[Π]Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University in Prague

Abstract

TrEd — the general-purpose tree editor being used within the two leading dependency treebanking projects in Czech and Arabic — has recently been extended with the MorphoTrees annotation context. On the case of Arabic, the novel concept of turning unorganized sets of complex morphological analyses into intuitive hierarchies is presented. Efficiency of disambiguation of MorphoTrees is discussed, and so is the independence of the idea on the language and the implementation.

1 Introduction

Let us refer to other sources instead of providing herein the introduction to the multi-level annotation projects of Prague Dependency Treebank (Hajič et al., 2001) and Prague Arabic Dependency Treebank (Hajič et al., 2004, in this volume), which apply the theory of Functional Generative Description (Sgall et al., 1986) to newspaper and newswire texts of Czech and Arabic, respectively.

The indispensable annotation environment for both of these projects is the TrEd tree editor written in Perl by Petr Pajas.[1] It is not only a fully programmable and customizable graphical user interface, but also an excellent suite of utilities for automated processing of the data (consistency checks and revising, batch conversions, search, difference evaluation, etc.).

The linguistic structures that get annotated as trees are commonly considered to belong to the domain of syntax,[2] and TrEd has been employed mainly for building analytical and tectogrammatical interpretations of languages. For morphological disambiguation, other tools were developed or adapted, which proved helpful for Czech, but rather uneasy for Arabic.

Our study will first outline the state-of-the-art style of Tim Buckwalter's morphological analyses (Buckwalter, 2002), being of the best computational models of Arabic morphology along with (Beesley, 2001) and (Kiraz, 2001). Conversion of this style, which in principle returns information on morphs rather than morphemes (Sproat, 1992), into an approximation of Functional Arabic Morphology (Smrž, in prep; Hajič et al., 2004, in this volume) sets the best grounds for MorphoTrees, and gives enough insight into the relevant issues of the morphological system.

2 Towards Functional Morphology

Arabic orthography prescribes to concatenate certain word forms with the preceding or the following ones, which makes the boundaries of syntactic units, **tokens**, obscure. While some tokens collapse into one compact **string**, others get delimited by white space, like in Figure 1.

1. *kay* *nu-ʿṭiy-a+ka* *ʾīyā+hu* — كي نعطيك إيّاه
   so-that *we-give+you* part.+*him*
2. *li+nu-ʿṭiy-a+ka* *ʾīyā+hu* — لنعطيك إيّاه
   so-that+we-give+you part.+*him*
3. *kay* *nu-ʿṭiy-a+ka+hu* — كي نعطيكه
   so-that *we-give+you+him*
4. *li+nu-ʿṭiy-a+ka+hu* — لنعطيكه
   so-that+we-give+you+him

Figure 1: Synonymous expressions of *so that we give you him*. Tying of tokens is conventional and dependent lexically (conjunctions *li+* ل vs. *kay* كي), perhaps morphologically (cases of pronouns), but not syntactically (yet, syntax does control the use of the particle *ʾīyā* إيّا (Fischer, 2001)).

Morphological analyzers recognize the strings and describe the readings of their components, but the partitioning is often not unique. Multiple interpretations of a string may imply different morphs and different tokens being involved. This inverse problem is exemplified in Figure 2.

The information returned by Buckwalter's morphological analyzer (Buckwalter, 2002) meets the format

```
(morph_composition) [lemma_ID]
   morph_1/tag_1 + ... + morph_n/tag_n
```

where the **morphs** group implicitly into the prefix, stem and suffix **segments**,[3] and the lemma identifies the semantically dominant morph, usually the stem. Morphs are labeled with **tags** giving them the feel that they must be morphemes, as pointed out in (Smrž, in prep).

Functional Arabic Morphology (Hajič et al., 2004) needs to re-group the morphs according to the syntactic units of the language, i.e. tokens of the syntactic trees like in (Žabokrtský and Smrž, 2003). Each token should have its grammatical categories determined completely.

[1]TrEd is licensed under the GNU General Public License and is available at `http://ckl.mff.cuni.cz/pajas/`.

[2]For morphosyntax and its derivational trees, see Section 5.

[3]Some authors use the term **segment** for what is called **morph** here, and allow the word forms to decompose to a series of or none prefixes and suffixes. Here, we will concentrate on **strings** and **tokens**, anyway, so this discrepancy is indifferent to us.

For instance, Buckwalter's morphology on the string `wbjAnbhA` وبجانبها meaning *and at her side* would yield

```
(wabijAnibihA) [jAnib_1]
   wa/CONJ + bi/PREP +
   jAnib/NOUN +
   i/CASE_DEF_GEN + hA/POSS_PRON_3FS
```

where the segments are now indicated by line-breaks. The tokens, however, read `wa` *wa+* و *and*, `bi` *bi+* بـ *at*, `jAnib+i` *ǧānib-i* جانب *side* and `hA` *+hā* ها *her*.

When re-grouping the morphs into tokens, the functional morphological information can be approximately derived from the original tags of morphs.[4] For every token, its sequence of tags (right column below) maps to a vector of values of grammatical categories. The tokens of our example will receive these converted, quasi-functional, positional tags (left column):

```
C---------  wa       CONJ
P---------  bi       PREP
N-------2R  jAnib+i  NOUN+CASE_DEF_GEN
S----3FS2-  hA       POSS_PRON_3FS
```

The positional notation starts with the major and minor part-of-speech, and proceeds through mood, voice, etc. up to person, gender, number, case, and definiteness. The values of the categories are unset, i.e. rendered with `-`, if either they are irrelevant for the particular part-of-speech (first position), or there is no explicit morph present, like no illusory gender and number in `jAnib+i`. On the contrary, categories may be implied in parallel, cf. suffixed possessive pronouns being treated as regular pronouns, but in functional genitive (position nine). Some values can only be set based on other knowledge, which is the case of formal reduced definiteness (position ten).[5]

The complete list of mappings from Buckwalter's tags to the quasi-functional positional ones is available from the authors.

3 The MorphoTrees Hierarchy

The classical concept of morphological analysis is, technically, to take an input substring of a discourse and produce a list of different strings, each of which represents a reading of the input in terms of the underlying lexical units and morphs, and some abstract labels revealing the process of derivation of the input from the lexical units.

The practice has been, at least in Arabic, that the output information is not organized any further. The different analyses are not clustered together according to their common features, and the output strings are linear in structure and need explicit parsing. It is very difficult for a human to interpret the analyses and to discriminate among them. For a machine, it is undefined how to compare the distance of two analyses, as they are naturally all unequal strings.

MorphoTrees is the idea of building effective and intuitive hierarchies over and among the input and output strings of morphological systems. It is especially interesting for Arabic and the Functional Morphology, but it is in no sense limited to either of these.

Let us consider the analyses of the string فهم. Some readings will interpret it as just one token related to the notion of *understanding*, but homonymous for several lexical units, each of which producing many distinct derivations written like this. Other readings will decompose the string to two co-occurring tokens, the first one, *fa+* فـ *so*, a conjunction, and the other one, هم, analyzed as a verb, noun or pronoun, each again ambiguous in functions. Now, follow this description in Figure 2.

The output strings of Tim Buckwalter's morphology were processed according to Section 2. The analyses and their elements were then merged into the five-level hierarchy, the leaves of which are the tokens and their tags as the atomic units, and the root being the input string, or generally an entity (some tree of discourse elements).

Rising from the leaves up, there is the level of lemmas of the lexical units, the level of non-vocalized standard orthographical forms, and the level of decomposition of the entity into a sequence of such forms, implying the number of tokens and their spelling.

We would like to stress that the hierarchy itself might define how to evaluate morphological taggers, lemmatizers and stemmers for Arabic, as their performance on the different levels and their combinations is of great interest.

In Figure 2, we also give analyses of افراد and اما to further clarify the point.

4 MorphoTrees Disambiguation in TrEd

Annotation of MorphoTrees rests in selecting the applicable sequence of leaves that analyze the entity in its context. An annotator could search the trees by sight, decoding the information for every possible analysis before coming across the right one ... Instead, MorphoTrees offer the option to restrict the subtrees and hide those leaves/branches that do not conform to the restrictions. Moreover, many restrictions may be applied automatically, and the decisions about the tree controlled in a very rapid and elegant way.

Two annotations are highlighted in Figure 2. For فهم, the annotator was expecting, from the context, the reading involving a conjunction. Upon pressing the shortcut `c`, the tree was restricted and the only one applicable leaf selected. However, the conjunction is a part of a two-token entity, and annotation of the second token must be performed. Automatically, all inherited restrictions were removed (the empty tag under هم), and the subtree unfolded again. The annotator moved to the lemma for the pronoun, and restricted its readings to the nominative by pressing `1`. There was no more decision needed, and annotation proceeded to the next entity.

Alternatively, the annotation could have been performed merely by typing `s1`. The restrictions would unambiguously lead to the nominative pronoun, and then, without

[4] Our current programming concern is to re-implement Tim Buckwalter's system, published under GNU GPL, so that it gives truly functional grammatical categories, and incorporate it into TrEd, which would enable the annotators to update the morphological lexicons and re-run the analyzer instantly during work.

[5] Similar positional tag notation has been used in various projects in the past, most notably the European Multext and Multext-East projects, for languages ranging from English to Czech to Hungarian.

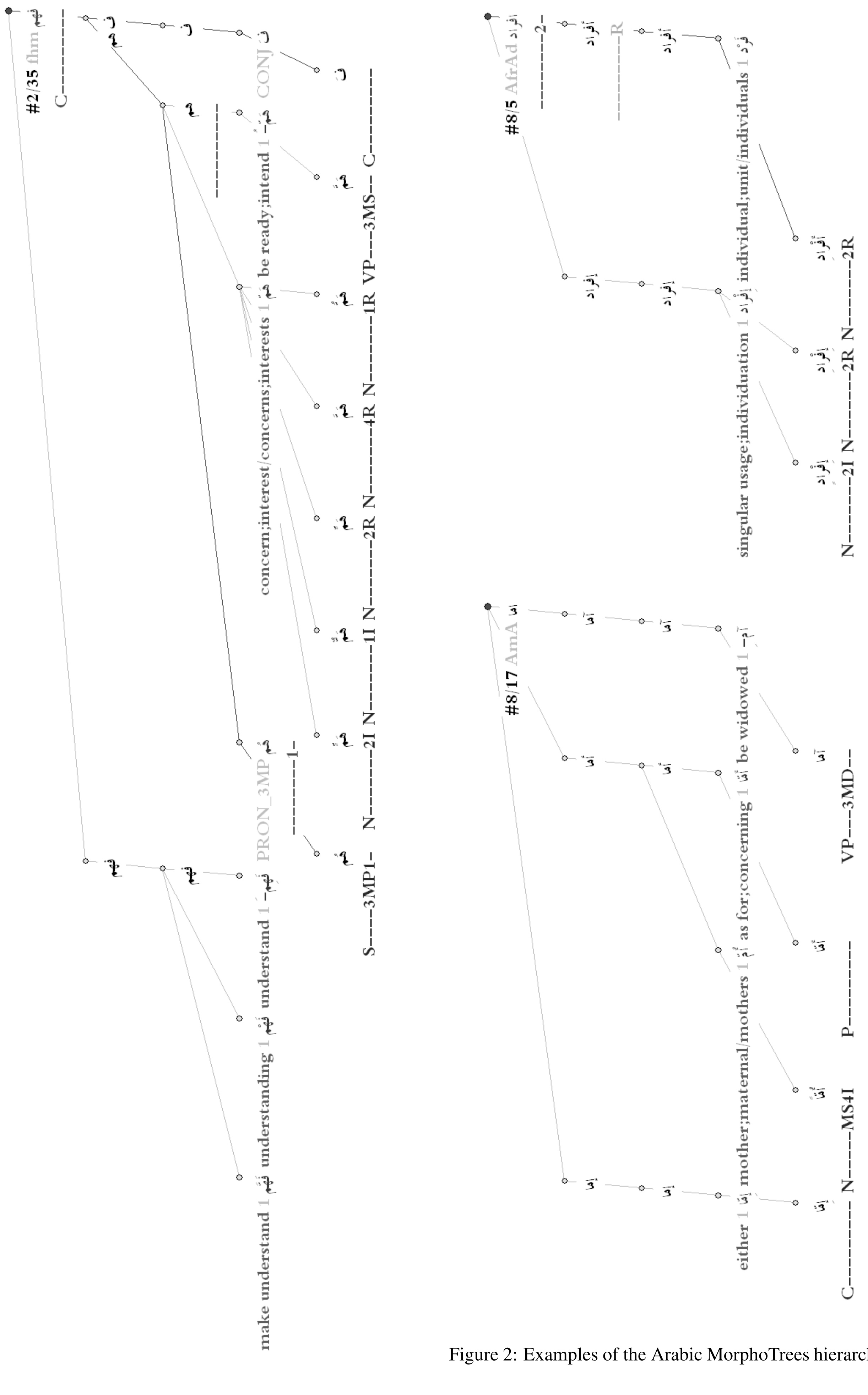

Figure 2: Examples of the Arabic MorphoTrees hierarchy.

human intervention, to the other token, the unambiguous conjunction. Let us note that the automatic decisions need no linguistic model, and yet they are very effective. Incorporating restrictions or forking preferences depending on the preceding annotations is just as simple.

The other annotation, افراد, illustrates clustering and the inheritance of restrictions. The annotator pressed `2` demanding a genitive. The decision moved one level lower, where spelling matters already. Both the branches are ambiguous between indefinite and reduced definiteness. The first reading holds, and `r` restricted it further, completing the annotation at the very moment.

5 Discussion and Conclusions

The levels of MorphoTrees are extensible internally (More decision steps for some languages?) and externally in both directions (Analyzed entity becoming a tree of discontiguous parts of a possible idiom? Leaves replaced with morphosyntactic trees of the morphs of the tokens?) and the concept brings a general view of many related problems.

In Arabic, whose MorphoTrees analyses get on average 7.9 tokens per entity and 1.4 partitions per entity, restrictions improve the speed of annotation incredibly.

Acknowledgements

The research described herein has been supported by the Ministry of Education of the Czech Republic, projects LN00A063 and MSM113200006.

Arabic script displays were typeset using the ArabTEX package for TEX and LATEX by Prof. Dr. Klaus Lagally of the University of Stuttgart.

References

Kenneth R. Beesley. 2001. Finite-State Morphological Analysis and Generation of Arabic at Xerox Research: Status and Plans in 2001. In *EACL 2001 Workshop Proceedings on Arabic Language Processing: Status and Prospects*, pages 1–8, Toulouse, France, July 2001.

Tim Buckwalter. 2002. Buckwalter Arabic Morphological Analyzer Version 1.0. LDC catalog number LDC2002L49, ISBN 1-58563-257-0.

Wolfdietrich Fischer. 2001. *A Grammar of Classical Arabic*. Yale Language Series. Yale University Press, third revised edition. Translated by Jonathan Rodgers.

Jan Hajič, Barbora Hladká, and Petr Pajas. 2001. The Prague Dependency Treebank: Annotation Structure and Support. In *Proceeding of the IRCS Workshop on Linguistic Databases*, pages 105–114, Philadelphia, December 2001. University of Pennsylvania.

Jan Hajič, Otakar Smrž, Petr Zemánek, Jan Šnaidauf, and Emanuel Beška. 2004. Prague Arabic Dependency Treebank: Development in Data and Tools. In *NEMLAR 2004 Conference Proceedings*.

George Anton Kiraz. 2001. *Computational Nonlinear Morphology with Emphasis on Semitic Languages*. Studies in Natural Language Processing. Cambridge University Press.

Petr Sgall, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. D. Reidel & Academia, Dordrecht & Prague.

Otakar Smrž. in prep. *Functional Arabic Morphology. Formal System and Implementation*. Ph.D. thesis, Charles University in Prague.

Richard Sproat. 1992. *Morphology and Computation*. ACL–MIT Press Series in Natural Language Processing. MIT Press.

Zdeněk Žabokrtský and Otakar Smrž. 2003. Arabic Syntactic Trees: from Constituency to Dependency. In *EACL 2003 Conference Companion*, pages 183–186, Budapest, Hungary, April 2003.