LEARNING TO USE THE PRAGUE ARABIC DEPENDENCY TREEBANK

Otakar Smrž, Petr Pajas, Zdeněk Žabokrtský,
Jan Hajič, Jiří Mírovský, Petr Němec
Charles University in Prague,
Institute of Formal and Applied Linguistics

1. Introduction

Prague Arabic Dependency Treebank (PADT), recently published in its first version (Hajič et al. 2004a) by the Linguistic Data Consortium, is both a collection of multi-level linguistic annotations over Modern Standard Arabic, and a suite of unique software implementations designed for general use in Natural Language Processing.

The underlying theory of this resource is overviewed in (Hajič et al. 2004b). In the current paper, we focus rather on the practical aspects of using the PADT data and the computational tools in original research.

1.1 *Data survey*

The corpus of PADT 1.0 consists of morphologically and analytically annotated newswire texts of Modern Standard Arabic, which originate from Arabic Gigaword (Graff 2003) and partly overlap with the plain data of Penn Arabic Treebank, Part 1 (Maamouri et al. 2003) and Penn Arabic Treebank, Part 2 (Maamouri et al. 2004).

The rough survey of the annotations is given in Table 1. Data sets AFP, UMH and XIN come from the earlier period of the project when morphological annotations were not based on the MorphoTrees technology (cf. Subsection 2.1). Therefore, the files recording the process of morphological disambiguation of these data could not be

distributed. Still, the resulting morphological information is available in the analytical files, along with the analytical annotations.

The other data sets, namely ALH, ANN and XIA, are full-fledged already and provide files of three different types — non-annotated text, MorphoTrees annotations, and analytical annotations. Information from the morphological level is also, as a prerequisite, propagated into the analytical level. Not all the data are processed on both levels, though.

Data	**[A] Tokens**	**[M]**	**Original Data Provider**	**News Period**
AFP	13 000	—	Agence France Presse	2000 / VII
UMH	38 500	—	Ummah Press Service	2002 / I–III
XIN	13 500	—	Xinhua News Agency	2003 / V
ALH	10 000	73 500	Al-Hayat News Agency	2001 / IX
ANN	12 500	25 500	An-Nahar News Agency	2002 / XI
XIA	26 500	49 500	Xinhua News Agency	2003 / V
113 500 Analytical level **148 000** MorphoTrees			TrEd Netgraph Oraculum Encode::Arabic **software + documentation**	

Table 1: Survey of the contents of the Prague Arabic Dependency Treebank 1.0. Columns [A] and [M] represent the number of syntactic units, i.e. tokens, for analytical level and MorphoTrees, respectively.

1.2 *Annotation environment*

The indispensable annotation environment for this and various other treebanking projects is the TrEd tree editor (Hajič et al. 2001) written in Perl/Tk. It is not only a fully programmable and customizable graphical user interface, but also an excellent suite of utilities for automated, optionally parallel, processing of the data (consistency checks and revising, batch conversions, search, difference evaluation, etc.).

TrEd is documented on http://ckl.mff.cuni.cz/pajas/tred/. We will explore some of its features in Subsection 4.2.

1.3 *Treebank search engines*

Netgraph (Mírovský & Ondruška 2002) is a client–server application for efficient searching in treebanks. Unlike TrEd, it provides the user with an easy-to-learn graphical query language that does not presume

any programming skills. The client application is implemented in Java, and is available on http://quest.ms.mff.cuni.cz/netgraph/.

Oraculum (Ljubopytnov et al. 2002) supports linguistically even more expressive queries, and operates through a sophisticated web browser interface, which is now being ported to Arabic.

1.4 *Other tools*

Next to several other linguistically significant solutions (cf. Section 5), there is the Encode::Arabic module (Smrž 2003) for Perl that supports miscellaneous modes of processing of the non-trivial, yet ingenious ArabTeX encoding notation of the Arabic script and/or its phonetic transcription (Lagally 2004). Encode::Arabic covers the Buckwalter transliteration, too.

2. **Data Structures**

The PADT annotations are distributed as UTF-8 encoded files in the FS format, which is documented on TrEd's website. TrEd and the array of associated tools and libraries provide options for converting these data into several XML-compliant formats, and vice versa. TrEd's graphical renderings can be printed as PostScript, PDF, or image files.

If independent data processing is desired, the files can best be accessed using the Fslib module for Perl, which is available in the distribution along with many other modules and scripts serving for data flow management, migration of annotations, updating and quality checking, difference evaluation or execution of systematic revisions.

The non-annotated textual data are provided in the original XML format of the Arabic Gigaword corpus.

2.1 *Functional morphology & MorphoTrees*

The morphological annotations of PADT used to directly employ the information produced by Buckwalter Arabic Morphological Analyzer (Buckwalter 2002). With the introduction of Functional Arabic Morphology (Smrž in prep.), all morphological tags were mapped as closely as possible into the current positional notation representing individual grammatical categories in separate columns.

The new type of annotations required a different disambiguation tool, and MorphoTrees (Smrž & Pajas 2004) came into existence, implemented as an annotation context for TrEd.

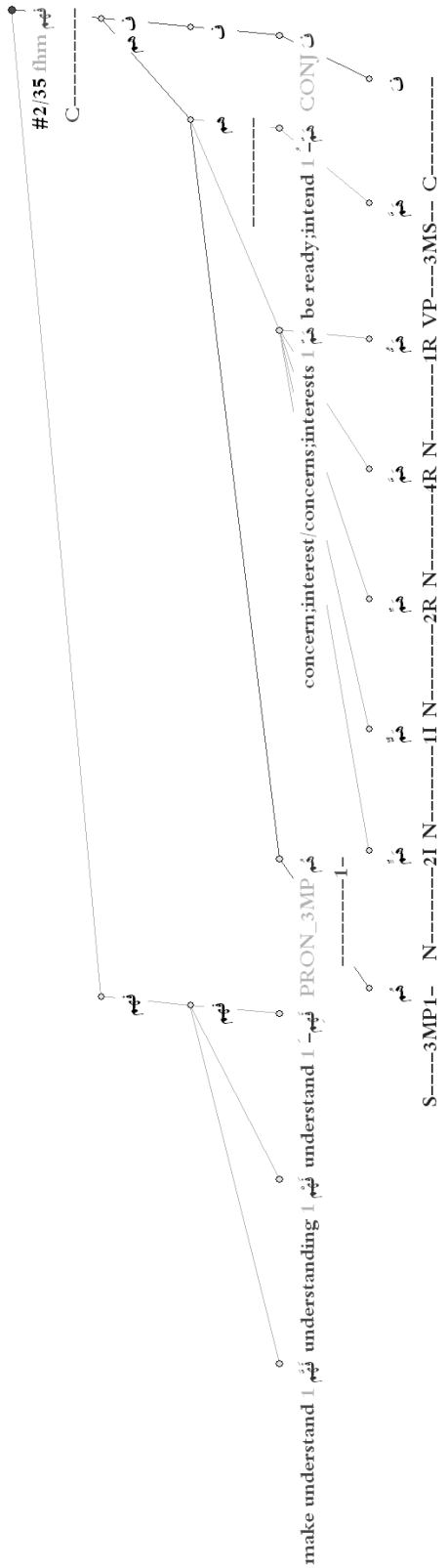

Figure 1 (top / left):
The hierarchy of MorphoTrees and their annotation using restrictions (cf. Smrž & Pajas 2004).

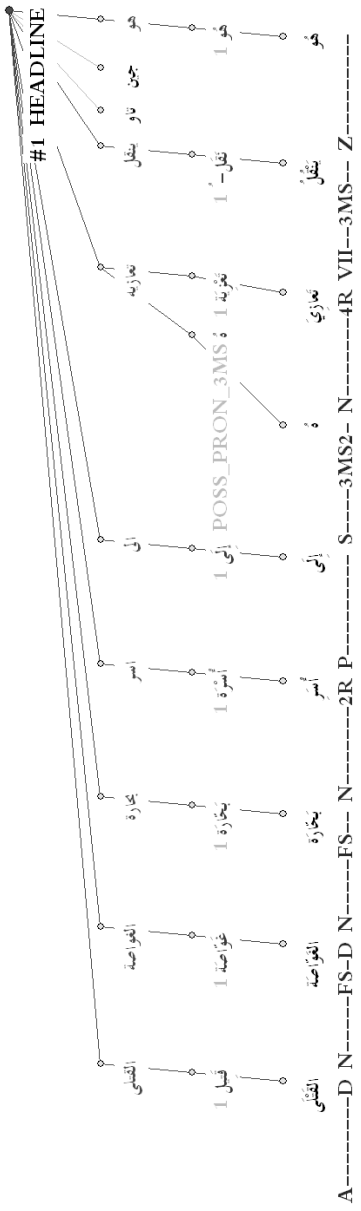

Figure 2 (bottom / right):
View of annotated paragraph. Note the levels of distinct information.

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 is in no sense limited to either of these.

Figure 1 illustrates how MorphoTrees organize the morphological information/analyses into a multi-level hierarchy. The leaves of these trees are the imaginable tokens with their tags as the atomic units, and the root is the input string being analyzed, 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 orthographic forms, and the level of decomposition of the entity into a sequence of such forms, implying the number of tokens and their spelling.

As a convenient extension, the overall solutions of the annotations can also be viewed in a similar hierarchical structure. An example of such a paragraph tree is given in Figure 2.

2.2 *Analytical dependency trees*

Analytical annotations represent the surface syntax of the language in the dependency formalism outlined in (Hajič et al. 2004b). They provide a link from morphology to tectogrammatics — the level of linguistic meaning — of the Functional Generative Description theory (cf. Sgall et al. 2004).

Analytical level is modeled with dependency trees whose nodes map, one to one, to the tokens resulting from the morphological analysis and tokenization, and whose roots group the nodes according to the division of the discourse into sentences or paragraphs.

Edges in the trees establish/reconstruct syntactic relations between the governor and the dependent, or rather, the whole subtree under and including the dependent. The nature of the government is expressed by the analytical functions of the nodes being linked.

In addition to this strict dependency structure, information of other kinds and character can be captured in the trees, while computational procedures for inferring any complementary information can be implemented independently of data. In TrEd, resolution of grammatical correference is automated in this manner. Identifying resumptive pronouns and deverbal inner objects by themselves is enough for some algorithm to find their grammatical counterparts and render these pairs.

Figure 3 (very right):
Analytical tree featuring advanced phenomena like ellipsis of another predicate, deverbal inner objects in adverbial function, or composite auxiliary elements. Note the labels [ExD] (on otherwise coordinative expression), [Adv_Msd], [AuxY] / [AuxP] (compound preposition) or [AuxY] / [ExD], respectively.

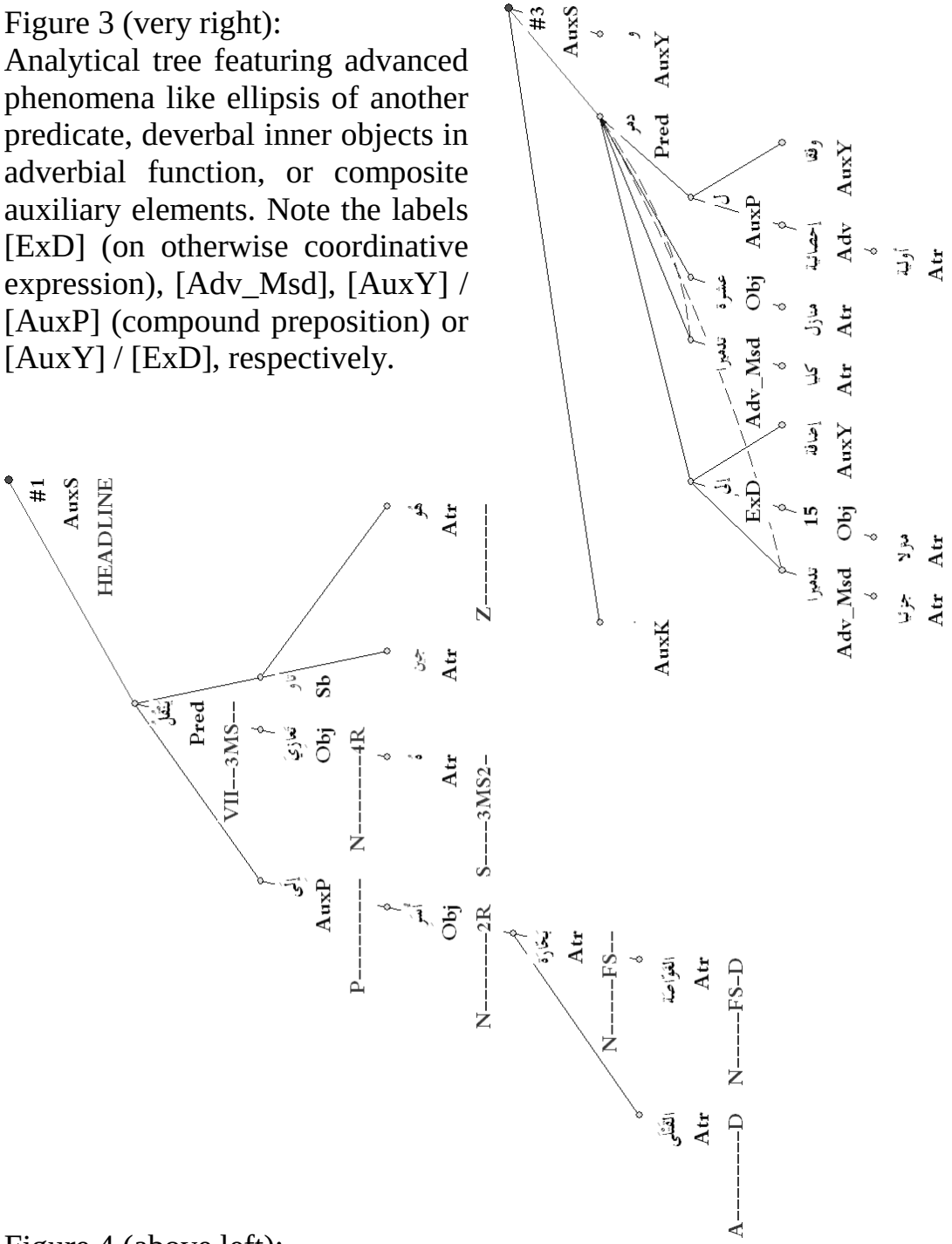

Figure 4 (above left):
Analytical representation of the sentence of Figure 2, with displayed morphological tags. Note the topology and functions of the predicate and its participants (subject, direct and indirect objects), and consider differences among the distinct attributive modifications.

In Figure 3, the instances of such non-dependency relations are shown with dashed arcs. Nonetheless, one might begin with Figure 4 for a more elementary example of an analytical tree.

3. Installation and General Setup

PADT 1.0 is distributed by Linguistic Data Consortium, University of Pennsylvania, http://www.ldc.upenn.edu/. The PADT project has its own website, http://ckl.mff.cuni.cz/padt/, where the data and the tools are documented in detail, and from where updates and extensions to the distribution are available.

User's installation should start with TrEd / Perl, and might proceed with downloading the Netgraph client / Java. The software applications are platform independent, and there is only little difficulty involved in setting things up. Installation of the data management scripts and modules or the CVS repositories for the FS annotation files is optional.

In order to search PADT with Netgraph, the client application must connect to a server accessing the data. Users are welcome to register with our Netgraph server, even though servers can also be run locally.

4. The Quest for IMPROPER ANNEXATION

Let us face the annotated data. Typically, linguists would like to search for a particular phenomenon in the language, evaluate it, contrast it with some other phenomena, consider the contexts of usage, etc.

The example case that we will explore in this section is the IMPROPER ANNEXATION in Arabic. A condensed definition of this phenomenon might not be precise — and we will not attempt it. Instead, we will pronounce and eventually refine our intuition that improper annexation is a genitive construction whose first term is an adjective, and whose second term is a [definite] noun (cf. for instance Schulz 2004:131–133,140,149).

We will, of course, use the treebank in order to test and improve the description of this notion. More importantly, we will learn about the applicability of PADT and its tools, and about some limitations.

4.1 *Querying PADT with Netgraph*

A query in Netgraph is a generalized subtree having the properties of the desired treebank structures specified as attributes of its individual nodes or edges. Queries can be created interactively through a graphical

interface, or equivalently, they can be linearized in a bracketing-style notation, which we will use here.

```
[tag=A?????????]
(
 [tag=N?????????,afun=Atr])
```

Figure 5: Netgraph query for the analytical level — a simple relation.

The example query in Figure 5 will return all occurrences of adjectives that have an attributive noun as one of its children. Such a relation is weaker than what improper annexation qualifies like. In particular, it ignores any constraints on word order, mutual distance, grammatical case and definiteness that we expect from a genitive construction. Anyway, it is just fine to ask Netgraph again and more specifically, adding some attributes to the nodes and listing the acceptable combinations of morphological categories in the tags. This gradual ruling out of irrelevant solutions is a helpful practice.

Netgraph queries need not concern the analytical level only. The structures in MorphoTrees can be investigated as well. Consider the query of Figure 6, which says: look for the paragraph trees, i.e. those whose root (_depth=0) is of type ‘paragraph’, in which we are interested in two immediately succeeding token nodes on the lowest level (_depth=3) such that the first one is a non-indefinite adjective and the second one is a non-indefinite noun either certainly in genitive, or with the value for case unset. Recall Figure 2 for better visualization.

```
[type=paragraph,_depth=0]
(
 [_transitive=true,_depth=3,
  _name=N1,
  type=token_node,
  tag=A????????C|A????????D|A????????R|A????????-]
 ,
 [_transitive=true,_depth=3,
  ord={N1.ord}+3,
  type=token_node,
  tag=N????????2C|N????????2D|N????????2R|N????????2-|
      N???????-C|N???????-D|N???????-R|N???????--])
```

Figure 6: Netgraph query for improper annexation in MorphoTrees.

Upon submitting this query to the server, we receive much more precise tips of what improper annexation could be. But when browsing through the results in Netgraph and trying to determine which of these are and which are not the appropriate cases, one may usually not see enough context of the surrounding paragraphs, and may not export the information in a very flexible way in order to process it further. Neither may the data be edited directly, if one is supposed to make corrections based on the search. How do we meet such requirements, then?

4.2 *Searching and viewing in TrEd*

TrEd, even in its graphical annotation mode, can work with filelists, by which we define the extent of the corpus where search operations are to take place. Besides the obligatory menu item ‘Node > Find ...’ by its attributes, there is the function ‘User-defined > Perl-Eval’ that executes a given Perl code in the current environment of TrEd’s data structures.

```
ChangingFile(0);            ## $this represents the current node

do {

  if ($this->root()->{'type'} eq 'paragraph') {

    $prev = undef;

    while ($this = $this->following()) {

      if ($this->{'type'} eq 'token_node') {

        if (defined $prev
                and $prev->{'tag'} =~ /^A........[CDR-]$/
                and $this->{'tag'} =~ /^N.......[2-][CDR-]$/
                and $this->{'ord'} == $prev->{'ord'} + 3) {

          return;
        }

        $prev = $this;
      }
    }
  }
}
while NextTree() || NextFile();
```

Figure 7: TrEd evaluation code in Perl, equal to the query of Figure 6.

The program in Figure 7 keeps iterating over the MorphoTrees data until the configuration of nodes discussed with Figure 6 is encountered. Then, the control returns to TrEd, which sets the cursor to the newly found occurrence of the hypothesized improper annexation.

The program in Figure 8 is designed for the analytical level, where the dependency information, rather than immediate adjacency, can be exploited. The algorithm carefully finds the head of the genitive construction even if its tail actually consists of multiple genitives in (hierarchical) coordination or apposition (cf. Figure 9, ex. E). Plus, there are constraints on the morphological tags of the nodes in question, relaxed a little with respect to the tagset of the former disambiguation.

```
ChangingFile(0);            ## $this represents the current node

do {

  while ($this = $this->following()) {

    if ($this->{'afun'} eq 'Atr' and
        $this->{'tag'} =~ /^N.......[23-][CDRX-]$/) {

      $head = $this;

      $head = $head->parent()
              while $head->{'parallel'} =~ /^(?:Co|Ap)$/;

      $head = $head->parent();

      return  if $head->{'tag'} =~ /^A........[CDRX-]$/;
    }
  }
}
while NextTree() || NextFile();
```

Figure 8: TrEd evaluation code for finding improper annexation on the analytical level. Note how coordination/apposition nodes between the two parts of the genitive construction are treated. Values 3 and X in the tags reflect some systematic ambiguity present in the old data sets.

It might be clear by now that this powerful mechanism of computing with trees can be abstracted from, and that the return instruction can be replaced with, say, printing out the current node's address and some significant attributes of its neighbors, or with code

for complex restructuring, or with simple counting. In fact, there are two important modifications of TrEd, named `btred` and `ntred`, with which almost every automatic processing, including searching, is done very quickly and conveniently. Please, consult the documentation.

4.3 *Improper annexation*

Having applied the criteria of Figure 7 and Figure 8 on our treebank data, we certainly did not obtain only improper annexations! How can we tell? And why have we not come up with the right kind of queries?

Let us refer for the answer to the first question to e.g. (Schulz 2004) or (Badawi et al. 2004). There are crucial semantic distinctions to make as to whether the adjectival head of the genitive construction logically qualifies the dependent noun, or whether this relation is reversed. Such information is neither present in morphology, nor in analytical syntax.

A.	*biṭāqatu*	*safarin*	*ʾuḥādīyatu*	*al-ittiǧāhi*	a one-way ticket
	N-----FS1R	N-------2I	A-----FS1R	N-------2D	
B.	*d̲āta*	*maʿāyīra*	*ʿāliyati*	*al-ǧawdati*	(being) of high-quality standards
	N-------4R	N-------2I	A-----FS2R	N-----FS2D	
C.	*bayna*	*ad-duwali*	*ad-dāʾimati*	*al-ʿuḍwīyati*	among the permanent member states
	P---------	N-------2D	A-----FS2C	N-----FS2D	
D.	*li-taṭbīqi*	*al-qawānīni*	*al-marʿīyati*	*al-ʾiǧrāʾi*	for application of the implementable laws
	P---------	N-------2D	A-----FS2C	N-------2D	
	N-------2R				
E.	*ʾinnahu*	*kāmilu*	*aṯ-ṯiqati*	*wa-al-maqdirati*	(indeed) he is of complete confidence and competence
	F---------	A-------1R	N-----FS2D	C---------	
	S----3MS4-			N-----FS2D	
F.	*al-ʿumlātu*	*ġayru*	*mustaqirrati*	*al-qīmati*	(these) currencies are of unstable value
	N-----FP1D	FN------1R	A-----FS2R	N-----FS2D	

O.	*li-al-ḥukūmati al-muntahiyati wilāyatuhā*	for the government whose term is ending
P.	*fī al-manāṭiqi al-ḫamsi as-sābiqi ḏikruhā*	in the five areas mentioned previously
Q.	*al-maʿūnāti al-bāliġata zinatuhā 20 ṭunnan*	the aid whose weight reaches 20 tons
R.	*bi-šaʾni al-intiḫābāti ar-riʾāsīyati al-muqarrari ʾiǧrāʾuhā fī rabīʿi 2004*	concerning the presidential elections that are agreed to take place in spring 2004

Figure 9: Contrasting improper annexation (examples A–F) with *na^{c}t sababī* (examples O–R). Note the patterns of definiteness or agreement in both of these phenomena (cf. e.g. Badawi et al. 2004:110–116).

On the other hand, our queries do include some looseness. Ideally, the values of the relevant morphological categories should all be set. Then, the definiteness values for the head of a genitive construction could only be R (reduced) or C (complex), as we exemplify in (Hajič et al. 2004b), and there would emerge other regularities that we could try to capture, or patterns that we could try to exclude.

In Figure 9, we give several examples of true improper annexation that we have found, and compare it with another phenomenon that partly invades the set of search results due to the unset case information of the nominatives therein.

Needless to say, preferring the recall of a query to its precision helps discover more inconsistencies or mistakes in annotation. The way we process the results in order to filter out false positives, like printing additional information, sorting and uniq-ing it, etc., is also important. In our current situation, roughly one out of six tips provided by the queries happened to be correctly classified as improper annexation.

Figure 10 summarizes the most interesting of these as observed in PADT — in its development version growing in size. Some of the phrases are rather idiomatic (cf. Wehr 1980), but what we notice is the actual freedom of expression and productivity of this linguistic construct. In the list, the heads of the annexations are lexicographically normalized, and the numbers in the rightmost column indicate the counts of occurrences within the treebank.

5. Applications and Prospects

The applicability of treebanks is very diverse. Not only, as we have just illustrated, can the annotated structures be studied in the educational or purely linguistic framework. The other prominent motivation is to use the data for machine-learning purposes, possibly aspiring to machine translation (cf. Čmejrek et al. 2004) or modeling of meaning.

In the course of the PADT project, we have developed systems for automatic morphological and analytical disambiguation, a.k.a. tagging and parsing (cf. Hajič et al. 2004b). This technology is going to be employed in the processing of the Arabic English Parallel News Part 1 (Ma 2004). Alternative automated annotation methods also come into question, like the parallel-corpus-based syntactic projection (Hwa et al. 2005) or the conversion of constituency annotations into dependencies (Žabokrtský & Smrž 2003; cf. Habash & Rambow 2004).

baṣalīyu	*aš-šakli*	onion-like, of onion shape	1
baʿīdu	*al-madā*	far-reaching, long-term	2
baʿīdu	*al-manāli*	unattainable, intangible	1
bāliġu	*al-ḥassāsīyati*	very sensitive	2
bāliġu	*at-taḫaṣṣuṣi*	very specialized	1
bāliġu	*aš-šiddati*	extremely strong, intense	1
bāliġu	*aḍ-ḍaʾālati*	negligible, insignificant	1
bāliġu	*at-taʿqīdi*	extremely complicated	1
bāliġu	*al-ʾahammīyati*	very important, of great importance	1
tāmmu	*aṣ-ṣunʿi*	ready-made, of completed production	1
maǧhūlu	*al-huwīyati*	unidentified, anonymous	1
ǧayyidu	*at-tanẓīmi*	well-organized	1
maḥdūdu	*ad-daḫli*	of limited income	2
ḥurru	*at-tadaffuqi*	unrestricted	1
maḥallīyu	*aṣ-ṣunʿi*	manufactured locally	1
muḫtalifu	*al-ʾaḥǧāmi*	of different sizes	1
ḏātīyu	*al-ḥukmi*	autonomous, self-governed	2
ḏātīyu	*ad-dafʿi*	self-propelled	2
rafīʿu	*al-mustawā*	high-level, of high level	18
murtafiʿu	*aṯ-ṯamani*	of increased price	1
šadīdu	*al-lahǧati*	of strong tone	1
šadīdu	*al-waṭʾati*	of intense pressure, oppressive	1
šāsiʿu	*al-ʾarāḍī*	of vast area	1
ṭawīlu	*al-ʾaǧali*	long-term, of long purpose	7
ṭawīlu	*al-ʾamadi*	long-term, long-extent, of long duration	1
ṭawīlu	*al-madā*	long-reach, long-distance	4
mutaʿaddidu	*at-taḫaṣṣuṣāti*	multidisciplinary, versatile	1
mutaʿaddidu	*al-ǧinsīyāti*	multi-national	4
mutaʿaddidu	*aṭ-ṭabaqāti*	multi-level, of multiple layers	1
mutaʿaddidu	*al-ʾaṭrāfi*	multi-lateral	3
ʿadīmu	*al-ʾaṭari*	ineffective	1
ʿālī	*al-mustawā*	of high level	2
ʿamīqu	*at-tafkīri*	thoughtful, of deep thought	2
maftūḥu	*as-saqfi*	open-roof	1
fāʾiqu	*al-qudrati*	highly capable, of outstanding potential	1
qaṣīru	*al-ʾaǧali*	short-term, of short purpose	4
qaṣīru	*an-naẓari*	short-sighted	1
munqaṭiʿu	*an-naẓīri*	incomparable, unparalleled	1
mutaqallibu	*al-ʾārāʾi*	wishy-washy, of wavering opinion	1
mutaqallibu	*al-mizāǧi*	moody	1
qawīyu	*aš-šakīmati*	active, energetic, vigorous	1
qawīyu	*al-muqāwamati*	of strong resistance, strongly opposing	1
kabīru	*at-taʾṯīri*	of big influence	1
kabīru	*as-sinni*	elderly, of elder age	3
kaṯīfu	*as-sukkāni*	of dense population	1
munamnamu	*al-ḥaǧmi*	miniature, of miniature size	1
mutanāhī	*aṣ-ṣiġari*	extremely tiny	1
mutanawwiʿu	*al-ʾaškāli*	of varied types, of various shapes	2
mahīḍu	*al-ǧanāḥi*	of broken wings	1
waṯīqu	*al-irtibāṭi*	closely related	1
waṯīqu	*aṣ-ṣilati*	firmly linked	2
wāsiʿu	*al-iṭṭilāʿi*	thoroughly informed	2
wāsiʿu	*an-niṭāqi*	far-reaching, large-scale	3
mutawāḍiʿu	*aḏ-ḏawqi*	of modest taste	1

Figure 10: Selected occurrences of improper annexation found on either level of the treebank.

We would as well like to implement algorithms for detection of inconsistencies and errors in the annotations (cf. Dickinson & Meurers 2003). The PADT website will offer any eventual updates. The current distribution already includes scripts for safe and maximally efficient migration of annotations if some data need to be synchronized and the changes propagated across the levels of description.

6. Conclusion

We have tried to give a practical introduction to the Prague Arabic Dependency Treebank project, with emphasis on PADT 1.0 available to researchers worldwide.

Having described the essential data structures in the treebank, we chose to search for and explore a particular linguistic phenomenon. We demonstrated the methodology for posing queries, and outlined how the information in the treebank might be processed in the general case.

We have presented and discussed the most noteworthy instances of improper annexation in Arabic that we found in the treebank using this methodology. This is a significant result by itself, and would be extremely hard to achieve without the kind of annotations the treebank provides. We would like to invite others to try their own queries.

Treebanking entails many challenging tasks, and we continue to approach them, as well as to improve the existing solutions.

7. Acknowledgements

The research described herein was supported by the Ministry of Education of the Czech Republic through projects LN00A063 and MSM113200006, and continues with the support from the Grant Agency of Charles University in Prague, project 207-10/203333. At the time of writing this paper, one of the authors was a grantee of the Fulbright-Masaryk Fellowship awarded by the Fulbright Commission in the Czech Republic.

The ‘quest for improper annexation’ was first suggested by Tim Buckwalter, while Iveta Kouřilová helped us with understanding and presenting the topic. We would like to thank them very much, too.

REFERENCES

Badawi, Elsaid & Carter, Mike G. & Gully, Adrian. 2004. *Modern Written Arabic: A Comprehensive Grammar*. Routledge, London.

Buckwalter, Tim. 2002. *Buckwalter Arabic Morphological Analyzer Version 1.0*. LDC catalog number LDC2002L49, ISBN 1-58563-257-0. Linguistic Data Consortium, University of Pennsylvania.

Čmejrek, Martin & Cuřín, Jan & Havelka, Jiří. 2004. "Prague Czech-English Dependency Treebank: Any Hopes for a Common Annotation Scheme?". *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, 47–54. Boston.

Dickinson, Markus & Meurers, W. Detmar. 2003. "Detecting Inconsistencies in Treebanks". *Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT 2003)*. Växjö.

Graff, David. 2003. *Arabic Gigaword*. LDC catalog number LDC2003T12, ISBN 1-58563-271-6. Linguistic Data Consortium, University of Pennsylvania.

Habash, Nizar & Rambow, Owen. 2004. "Extracting a Tree Adjoining Grammar from the Penn Arabic Treebank". *Proceedings of Traitement Automatique du Langage Naturel (TALN-04)*. Fez.

Hajič, Jan & Hladká, Barbora & Pajas, Petr. 2001. "The Prague Dependency Treebank: Annotation Structure and Support". *Proceedings of the IRCS Workshop on Linguistic Databases*, 105–114. University of Pennsylvania.

Hajič, Jan & Smrž, Otakar & Zemánek, Petr & Pajas, Petr & Šnaidauf, Jan & Beška, Emanuel & Kráčmar, Jakub & Hassanová, Kamila. 2004a. *Prague Arabic Dependency Treebank 1.0*. LDC catalog number LDC2004T23, ISBN 1-58563-319-4. Linguistic Data Consortium, University of Pennsylvania.

Hajič, Jan & Smrž, Otakar & Zemánek, Petr & Šnaidauf, Jan & Beška, Emanuel. 2004b. "Prague Arabic Dependency Treebank: Development in Data and Tools". *Proceedings of the NEMLAR International Conference on Arabic Language Resources and Tools*, 110–117. Cairo.

Hwa, Rebecca & Resnik, Philip & Weinberg, Amy & Cabezas, Clara & Kolak, Okan. 2005. "Bootstrapping Parsers via Syntactic Projection across Parallel Texts". *Natural Language Engineering*, June 2005.

Lagally, Klaus. 2004. *ArabTeX: Typesetting Arabic and Hebrew*. User Manual Version 4.00, Fakultät Informatik, Universität Stuttgart.

Ljubopytnov, Vladimír & Němec, Petr & Pilátová, Michaela & Reschke, Jakub & Stuchl, Jan. 2002. "Oraculum, a System for Complex Linguistic Queries". *SOFSEM 2002 Student Research Forum*, 27–34.

Ma, Xiaoyi. 2004. *Arabic English Parallel News Part 1*. LDC catalog number LDC2004T18, ISBN 1-58563-310-0. Linguistic Data Consortium, University of Pennsylvania.

Maamouri, Mohamed & Bies, Ann & Jin, Hubert & Buckwalter, Tim. 2003. *Arabic Treebank: Part 1 v 2.0*. LDC catalog number LDC2003T06, ISBN 1-58563-261-9. Linguistic Data Consortium, University of Pennsylvania.

Maamouri, Mohamed & Bies, Ann & Buckwalter, Tim & Jin, Hubert. 2004. *Arabic Treebank: Part 2 v 2.0*. LDC catalog number LDC2004T02, ISBN 1-58563-282-1. Linguistic Data Consortium, University of Pennsylvania.

Mírovský, Jiří & Ondruška, Roman. 2002. "Netgraph System: Searching through the Prague Dependency Treebank". *Prague Bulletin of Mathematical Linguistics*, (77):101–104.

Schulz, Eckehard. 2004. *A Student Grammar of Modern Standard Arabic*. Cambridge University Press.

Sgall, Petr & Panevová, Jarmila & Hajičová, Eva. 2004. "Deep Syntactic Annotation: Tectogrammatical Representation and Beyond". *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, 32–38. Boston.

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

Smrž, Otakar. 2003. *Encode::Arabic*. Programming module. Comprehensive Perl Archive Network, http://search.cpan.org/dist/Encode-Arabic/.

Smrž, Otakar & Pajas, Petr. 2004. "MorphoTrees of Arabic and Their Annotation in the TrEd Environment". *Proceedings of the NEMLAR International Conference on Arabic Language Resources and Tools*, 38–41. Cairo.

Wehr, Hans. 1980. *A Dictionary of Modern Written Arabic. Arabic–English*. Spoken Language Service, New York.

Žabokrtský, Zdeněk & Smrž, Otakar. 2003. "Arabic Syntactic Trees: from Constituency to Dependency". *EACL 2003 Conference Companion*, 183–186. Budapest.