BUILDING AND EXPLOITING SYNTACTICALLY ANNOTATED CORPORA

BUILDING AND EXPLOITING SYNTACTICALLY ANNOTATED CORPORA

Edited by
ANNE ABEILLE
University of Paris

Kluwer Academic Publishers
Boston/Dordrecht/London

Contents

List of Figures

List of Tables

Chapter 1

BUILDING A TURKISH TREEBANK

Kemal Oflazer
Faculty of Engineering and Natural Sciences
Sabancı University
İstanbul, Turkey
oflazer@sabanciuniv.edu

Bilge Say
Informatics Institute
Middle East Technical University
Ankara, Turkey
bsay@ii.metu.edu.tr

Dilek Zeynep Hakkani-Tür
Faculty of Engineering and Natural Sciences
Sabancı University
İstanbul, Turkey
hakkani@sabanciuniv.edu

Gökhan Tür
Department of Computer Engineering
Bilkent University
Ankara, Turkey
tur@cs.bilkent.edu.tr

Abstract: We present the issues that we have encountered in designing a treebank architecture for Turkish along with rationale for the choices we have made for various representation schemes. In the resulting representation, the information encoded in the complex agglutinative word structures are represented as a sequence of inflectional groups separated by derivational boundaries. The syntactic relations are encoded as labeled dependency relations among segments of lexical items marked by derivation boundaries. Our current work involves refining a set of treebank annotation guidelines and developing a sophisticated annotation tool with an extendible plug-in architecture for morphological analysis, morphological disambiguation and syntactic annotation disambiguation.

Keywords: Treebanks, Dependency Syntax, Turkish, Agglutinative Languages

Introduction

In the last few years, treebank corpora such as the Penn Treebank [12, 13] or the Prague Dependency Treebank [2] have become a crucial resource for building and evaluating natural language processing tools and applications. Although the compilation of such structurally annotated corpora is time-consuming and expensive, the eventual benefits outweigh this initial cost. With a set of future applications in mind, we have undertaken the design of a treebank corpus architecture for Turkish, which we believe encodes the lexical and structural information relevant to Turkish. In this chapter we present the issues that we have encountered in designing a treebank for Turkish along with rationale for the representation choices we have made. In the resulting representation, the information encoded in the complex agglutinative word structures is represented as a sequence of inflectional groups separated by derivational boundaries. A tagset reduction is not attempted as any such reduction leads to removal of potentially useful syntactic markers, especially in the encoding of derived forms. At the syntactic level, we have opted to just represent relationships between lexical items (or rather, inflectional groups) as dependency relations. The representation is extensible so that relations between lexical items can be further refined by augmenting syntactic relations using finer distinctions which are more semantic in nature.

1. TURKISH: MORPHOLOGY AND SYNTAX

Turkish is an Ural-Altaic language, having agglutinative word structures with productive inflectional and derivational processes. Derivational phenomena have rarely been addressed in designing tagsets, and in the context of Turkish, this may pose challenging issues, as the number of forms one can derive from a root form may be in the millions [8].

Turkish word forms consist of morphemes concatenated to a root morpheme or to other morphemes, much like beads on a string. Except for a very few exceptional cases, the surface realizations of the morphemes are conditioned by various morphophonemic processes such as vowel harmony, vowel and consonant elisions. The morphotactics of word forms can be quite complex when multiple derivations are involved. For instance, the derived modifier `sağlamlaştırdığımızdaki`[1] would be represented as:[2]

```
sağlam+Adj^DB
      +Verb+Become^DB
      +Verb+Caus+Pos^DB
      +Adj+PastPart+P1sg^DB
      +Noun+Zero+A3sg+Pnon+Loc^DB
      +Adj
```

Marking such a word as an adjective and ignoring anything that comes before the last part of speech would ignore the fact that the stem is also an adjective which may have syntactic relations with preceding words such as an adverbial modifier, or that there is an intermediate causative (hence transitive) verb which may have an object NP or a subject NP to its left.

A recent experiment that we conducted on about 250,000 Turkish words in news text revealed that there were over 6,000 distinct morphological feature combinations when root morphemes were ignored. Although this is less than the much larger numbers quoted by Hankamer who considered the generative capacity of the derivations, it is nevertheless much larger than the distinctions encoded by the tagsets of languages like English or French. What is important is not the size of the potential tagset, but rather

- the fact that there is no a priori limit on it as the next set of million words that one looks at may contain another 6,000 distinct feature combinations, and
- the nature of the derivational information.

On the syntax side, although Turkish has unmarked SOV constituent order, it is considered a free-constituent order language as all constituents including the verb, can move freely as demanded by the discourse context with very few syntactic constraints [4]. Case marking on nominal constituents usually indicates their syntactic role. Constituent order in embedded clauses is substantially more constrained but deviations from the default order, however infrequent, can still be found. Turkish is also a pro-drop language, as the subject, if necessary, can be elided and recovered from the agreement markers on the verb. Within noun phrases, there is a loose order with specifiers preceding modifiers, but within each group, order (e.g., between cardinal and attributive modifiers) is mainly determined by which aspect is to be emphasized. For instance the Turkish equivalents of *two young men* and *young two men* are both possible: the former being the neutral case or the case where youth is emphasized, while the latter is the case where the cardinality is emphasized. A further but relatively minor complication is that various verbal adjuncts may intervene in well-defined positions within NPs causing discontinuous constituents.

2. WHAT INFORMATION NEEDS TO BE REPRESENTED?

We expect this treebank to be used by a wide variety of "consumers", ranging from linguists investigating morphological structure and distributions, syntactic structure, constituent order variations, to computational linguists extracting language models or evaluating parsers, etc. We would therefore employ an extendable multi-tier representation, so that any future extensions can be easily incorporated if necessary. Similar concerns have also been addressed in the French Treebank [1].

2.1 REPRESENTING MORPHOLOGICAL INFORMATION

At the lowest level we would like to represent three main aspects of a lexical item:

- The word itself, e.g., `evimdekiler`, (those in my house).
- The lexical structure, as a sequence of free and bound morphemes (including any morphophonological material elided on the surface, and meta symbols for relevant phonological categories), e.g.,

  `ev+Hm+DA+ki+lAr`

  (where for instance `D` represents a set of dental consonants, `H` a set of high-vowels and `A` represents the set of non-round front vowels, which are resolved to their surface realizations when the phonological context is taken into account.)
- The morphological features encoded by the word as a sequence of morphological and POS feature values all of which except the root are symbolic, e.g.,

`ev+Noun+A3sg+P1sg+Loc^DB+Adj^DB+Noun+Zero+A3pl+Pnon+Nom`

A point to note about this representation is that, information that is conveyed covertly by zero-morphemes that is not explicit in the lexical representation, is represented here. (e.g., if a plural marker is not present then the noun is singular hence `+A3sg` is the feature supplied even though there is no overt morpheme.) A comprehensive list of morphological feature symbols is given in Appendix A.

The first two components of the morphological information do not deserve any more details for the purposes of this presentation. The third component with its relation to lexical tag information needs to be detailed further.

The prevalence of productive derivational word forms brings a challenge to representing such information using a finite (and possibly reduced) tagset. The usual approaches to tagset design, typically assume that the morphological information associated with a word form can be encoded using a finite number of cryptically coded symbols from some set whose size ranges from few tens (e.g., Penn Treebank tag set [12]) to hundreds or even thousands (e.g., Prague Treebank tagset, [5, 2]). But, such a finite tagset approach for languages like Turkish inevitably leads to loss of information. The reason for this is that the morphological features of intermediate derivations can contain markers for syntactic relationships. Leaving out this information within a fixed-tagset scheme may prevent crucial syntactic information from being represented.

For these reasons we have decided not to compress in any way the morphological information associated with a Turkish word and represent such words as a sequence of *inflectional groups* (IGs hereafter), separated by `^DB`s denoting derivation boundaries. Thus a word would be represented in the following general form:

`root+Infl`$_1$`^DB+Infl`$_2$`^DB+`$\cdots$`^DB+Infl`$_n$

where `Infl`$_i$ denote relevant inflectional features including the part-of-speech for the root or any of the subsequent derived forms, if any. For instance, the derived modifier `sağlamlaştırdığımızdaki` (with the parse given earlier) would be represented by the 6 IGs:

1. `sağlam+Adj`
2. `+Verb+Become`
3. `+Verb+Caus+Pos`
4. `+Adj+PastPart+P1sg`
5. `+Noun+Zero+A3sg+Pnon+Loc`
6. `+Adj`

Note that the set of possible IGs is finite and these can be compactly coded into (cryptic) symbols, but we feel that apart from saving storage, such an encoding serves no real purpose while the resulting opaqueness prevents facilitated access to component features.

Although we have presented a novel way of looking at the lexical structure, the reader may have received the impression that words in Turkish have overly complicated structures with many IGs per word. The situation as indicated by various statistics actually indicate that this is really not the case. For instance the statistics presented in Table 1.1, compiled from about 850,000 word corpus of Turkish news text indicate that on the average the number of IG's per words is less than 2. Thus, for instance modelling each word uniformly with 2 IGs may be a very good approximation for statistical modeling [6].

Table 1.1 Parse and IG Statistics from a Turkish Corpus

	All tokens	**All but high frequency function words and and punctuation**
Morph. Parses per Token	1.76	1.93
IGs per Parse	1.38	1.48
% Tokens with single parse	55	45
% Parses with 1 IG	72	65
% Parses with 2 IGs	18	23
% Parses with 3 IGs	7	9
% Parses with > 3 IGs	3	3
Max Number of IGs in a parse	7	7
Distinct IGs ignoring roots	2448	

Turkish is also very rich in lexicalized and non-lexicalized collocations [16, 17]. The lexicalized collocations are much like what one would find in other languages. On the other hand, non-lexicalized collocations can be divided into two groups:

1. In the first group, we have compound and support verb formations where there are two or more lexical items the last of which is a verb. Even though the other components can themselves be inflected, they can be assumed to be fixed for the purposes of the collocation and the collocation assumes its inflectional features from the inflectional features of the last verb which itself may undergo any morphological derivation or inflection process. For instance, the idiomatic verb *kafa çek-* `(kafa+Noun+A3sg+Pnon+Nom`

çek+Verb+...) (literally, to pull head) means *to get drunk*, and these two tokens essentially behave together as far as syntax goes.[3]

2. The second group of non-lexicalized collocations involve full or partial duplication of verb, adjective or noun forms. For instance, the aorist marked verb sequence

   gelir gelmez (gel+Verb+Pos+Aor+A3sg gel+Verb+Neg+Aor+A3sg)

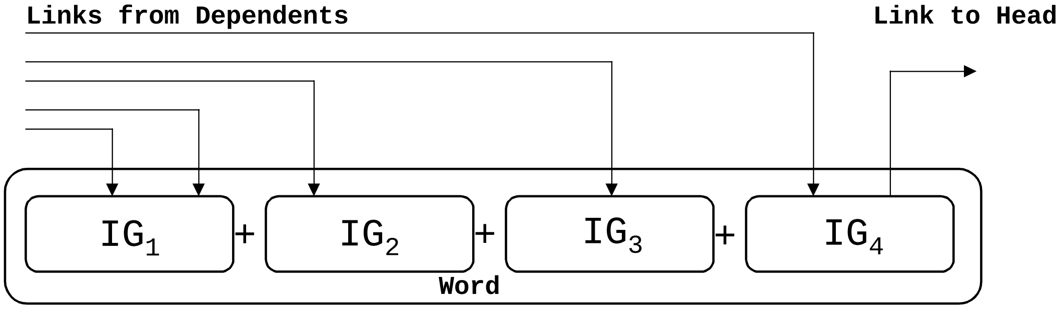
   actually functions as a temporal adverbial meaning *as soon as ... comes.* Note that these formations (usually involving full or partial reduplications of strings of the sort $\omega\ \omega$, $\omega\ x\ \omega$ or $\omega x\ \omega z$) are beyond the formal power of finite state mechanisms, hence are not dealt within the finite state morphological analyzer. (See Oflazer and Kuruöz [16] or Oflazer and Tür [17], for a list of such non-lexicalized collocations.)

2.2 REPRESENTING SYNTACTIC RELATIONS

We would like to represent syntactic relations between lexical items (actually between inflectional groups as we will see in a moment) using a simple dependency framework. Our arguments for this choice essentially parallels those of recent works on this topic [5, 2, 19, 3, 10]. Free constituent ordering and discontinuous phrases make use of constituent-based representations rather difficult and unnatural to employ. It is however possible to use constituency where it makes sense and bracket sequences of tokens to mark segments in the texts whose internal dependency structure would be of little interest. For instance, collocations, time–date expressions or multiword proper names (which incidentally do not follow Turkish noun phrase rules so have to be treated specially anyway) are examples whose internal structure is of little syntactic concern, and can be bracketed a priori as chunks and then related to other constituents. Such features have also been proposed for the French Treebank [1]. If necessary, any further constituent-based representation can be extracted from the dependency representation [11].

An interesting observation that we can make about Turkish is that, when a word is considered as a sequence of IGs, syntactic relation links only emanate from the last IG of a (dependent) word, and land on one of the IGs of the (head) word on the right (with minor exceptions), as exemplified in Figure 1.1. A second observation is that, (again with minor exceptions), the dependency links between the IGs, when drawn above the IG sequence, do not cross (although this is not a concern here).[4] Figure 1.3 shows a dependency tree for the following sentence

Figure 1.1 Links and Inflectional Groups



in Figure 1.2, laid on top of the words segmented along IG boundaries. Note for instance that, for the word *büyümesi* the previous two words link to its first (verbal) IG, while its 2nd IG (infinitive nominal) links to the final verb as subject.

Figure 1.2 Example Turkish Sentence

(1) Bu eski bahçe-de+ki
`bu(this)+Det` `eski(old)+Adj` `bahçe(garden)+A3sg+Pnon+Loc^DB+Adj`
The growth of the rose

gül-ün böyle
`gül(rose)+Noun+A3sg+Pnon+Gen` `böyle(like-this)+Adv`
like this in this old garden impressed everybody.

büyü+me-si
`büyü(grow)+Verb+Pos^DB+Noun+Inf+A3sg+P3sg+Nom`

herkes-i çok
`herkes(everybody)+Pron+A3sg+Pnon+Acc` `çok(very)+Adv`

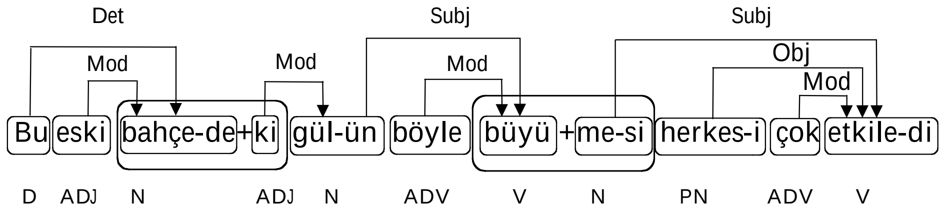
etkile-di.
`etkile(impress)+Verb+Pos+Past+A3sg`

The syntactic relations that we have currently opted to encode in our syntactic representation are the following:

1. Subject
2. Object
3. Modifier (adv./adj.)
4. Possessor
5. Classifier
6. Determiner
7. Dative Adjunct
8. Locative Adjunct
9. Ablative Adjunct
10. Instrumental Adjunct

Some of the relations above perhaps require some more clarification. *Object* is used to mark objects of verbs and the nominal complements of postpositions. A *classifier* is a nominal modifier in nominative case (as in *book cover*) while a *possessor* is a genitive case-marked nominal modifier. For verbal adjuncts, we indicate the syntactic relation with a marker paralleling the case marking though the semantic relation they encode is not only determined by the case marking but also the lexical semantics of the head noun and the verb they are attached to. For instance a dative adjunct can be a *goal*, a *destination*, a *beneficiary* or a *value carrier* in a transaction, or a *theme*, while an ablative adjunct may be *reason*, a *source* or a *theme*. Although we do not envision the use of such detailed relation labels at the outset, such distinctions can certainly be useful in training case-frame based transfer modules in machine translation systems to select the appropriate prepositions in English for instance.

Figure 1.3 Dependency structure for a sample Turkish Sentence



Last line shows the final POS for each word.

2.3 EXAMPLE OF A TREEBANK SENTENCE

In this section we present the detailed representation of a Turkish sentence in the treebank. Each sentence is represented by a sequence of attribute lists of the words involved, bracketed with tags `<S>` and `</S>`.[5] Figure 1.4 shows the treebank encoding for the sentence given earlier. Each word is bracketed by `<W>` and `</W>` tags. The `IX` denotes the number or index of the word. `LEM` denotes the lemma of the word, as one would find in a dictionary. For verbs, this would typically be an infinitive form, while for other word classes it would usually be the root

Figure 1.4 Sample treebank encoding a Turkish sentence

```
<S>
<W IX=1 LEM="bu" MORPH="bu" IG=[(1, "bu+Det")] REL=[(3,1,(DETERMINER)]>
Bu </W>

<W IX=2 LEM="eski"’ MORPH="eski" IG=[(1, "eski+Adj")]
REL=[3,1,(MODIFIER)]> eski> </W>

<W IX=3 LEM="bahçe" MORPH="bahçe+DA+ki" IG=[(1, "bahçe+A3sg+Pnon+Loc")
(2, "+Det")] REL=[4,1,(MODIFIER)]> bahçedeki </W>

<W IX=4 LEM="gül" MORPH="gül+nHn" IG=[(1,"gül+Noun+A3sg+Pnon+Gen")]
REL=[6,1,(SUBJECT)]> gülün </W>

<W IX=5 LEM="böyle" MORPH="böyle" IG=[(1,"böyle+Adv")]
REL=[6,1,(MODIFIER)]> böyle </W>

<W IX=6 LEM="büyümek" MORPH="büyü+mA+sH" IG=[(1,"büyü+Verb+Pos") (2,
"+Noun+Inf+A3sg+P3sg+Nom")] REL=[9,1,(SUBJECT)]> büyümesi </W>

<W IX=7 LEM="herkes" MORPH="herkes+yH" IG=[(1,"herkes+Pron+A3sg+Pnon+Acc")]
REL=[9,1,(OBJECT)]> herkesi </W>

<W IX=8 LEM="çok" MORPH="çok" IG=[(1,"çok+Adv’’)] REL=[9,1,(MODIFIER)]>
çok </W>

<W IX=9 LEM="etkilemek" MORPH="etkile+DH" IG=[(1,
"etkile+Verb+Pos+Past+A3sg")] REL=[]> etkiledi </W>

</S>
```

word itself. `MORPH` indicates the morphological structure of the word as a sequence of morphemes, essentially corresponding to the lexical form. The morphemes may involve meta-symbols (mentioned earlier) for indicating any phonological classes of symbols. `IG` is a list of pairs of an integer and an inflectional group. `REL` encodes the relationship of this word, as indicated by its last inflection group, to an inflectional group of some other word. The first component of `REL` is the index of a word, the second component is the number of the inflection group in that word that this word's last IG is linked to, and the third component is a list of relation labels for any possible syntactic (e.g., dative adjunct) and semantic (e.g., destination),relationships between the IGs involved. For example, the 4^{th} and 5^{th} words in the sentence are subject and and adverbial modifier, respectively, of the verb in the first IG of the 6^{th} word, while the 2^{nd} IG of the same word (6) is the subject of the main verb of the word 9. We have only used simple syntactic relation names in the example but more certainly can be added. For instance adjective modifiers can be further classified into attributive, cardinal, etc., while an object may further be marked as theme or patient, as discussed earlier.

A collocation would be represented by coalescing the information of individual components. For instance, the non-lexicalized collocation *gelir gelmez* and its adjunct

(2)	ev+e	gel+ir	gel+me+z
	`ev+Noun+A3sg+Pnon+Dat`	`gel+Verb+Pos+Aor+A3sg`	
		`gel+Verb+Neg+Aor+A3sg`	
	... as soon as ... comes to the house ...		

would be represented as

```
     ...
<W IX=5 LEM="ev" MORPH="ev+yA" IG=[(1,"ev+A3sg+Pnon+Dat")]],
    REL=[6,1,(DATIVE-ADJ,DEST)]> eve </W>

<W IX=6 LEM="gelmek" MORPH="gel+Hr gel+mA+z"
    IG=[(1, "gel+Verb+Pos")(2, "+Adv+AsSoonAs")],
    REL=[...]> gelir gelmez  </W>
    ...
```

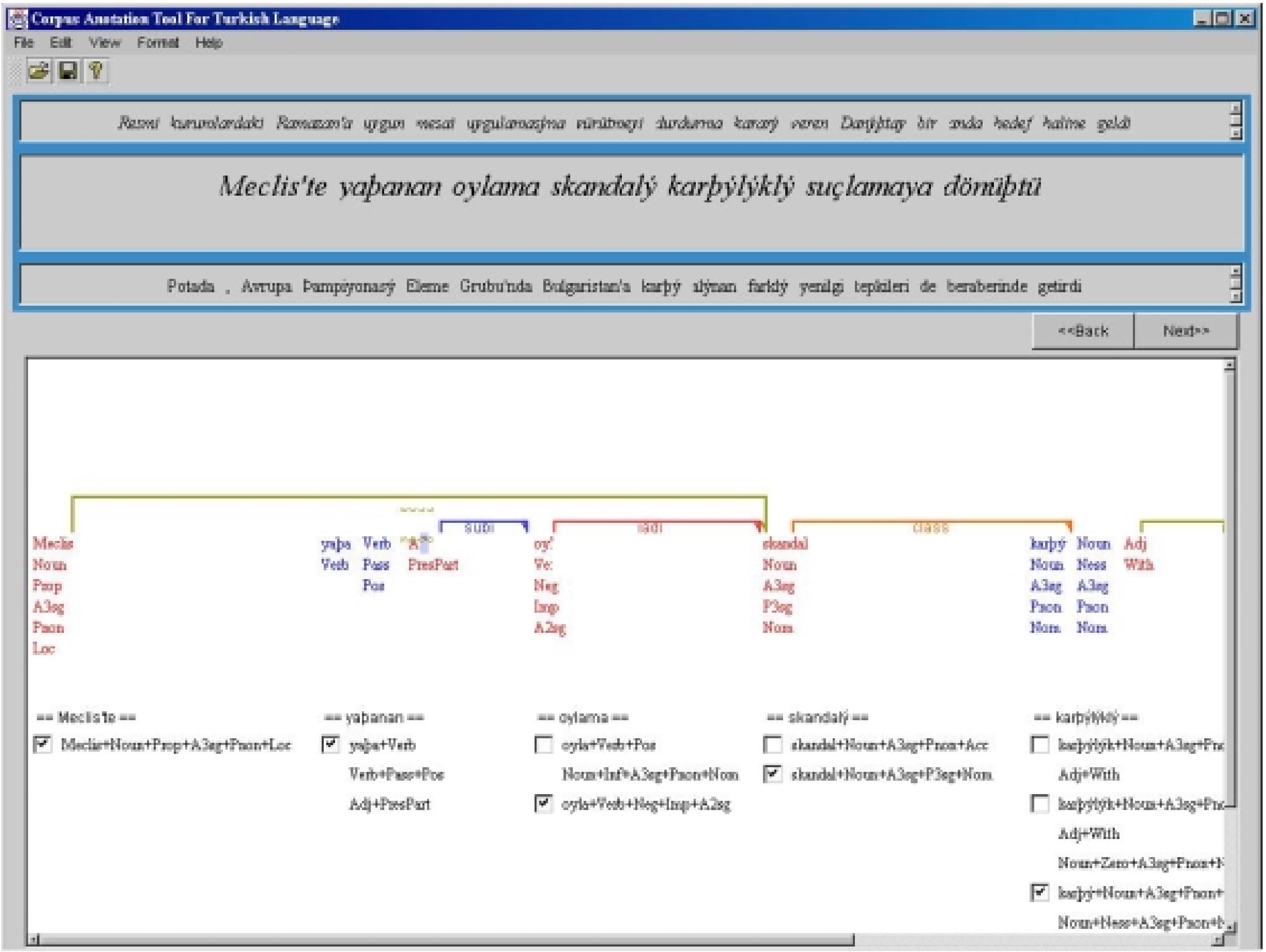
where it should be noted that the non-lexicalized collocation has been treated as derivational process and an adverbial IG `+Adv+AsSoonAs` has been created.

3. THE ANNOTATION TOOL

We have implemented a first version of treebank annotation tool that lets an annotator semi-automatically annotate a Turkish text. A snapshot of the user interface of this tool is given in Figure 1.5.

On the top, the annotator sees the sentence as text along with the previous and the next sentences, if any. The main window below contains the morphological analyses of the tokens with ambiguous analyses being listed vertically below the token. The annotator then performs a manual morphological disambiguation by selecting the appropriate analysis with a tick box.[6] The IGs of the selected analysis are then listed side by side, on the middle of the lower window, with the morphological features in an IG being listed vertically (see the entries above the rightmost word bracketed with ==). The annotator then proceeds with a drag and drop interaction, clicking on a source IG, starting a link and then drops the end of the link on the target IG. At this point a pop-up menu forces the

Figure 1.5 The user interface of the treebank annotation tool



Resmi kurumlardaki Ramazan'a uygun mesai uygulamasýna yürütmeyi durdurma kararý veren Danýþtay bir anda hedef haline geldi

Meclis'te yaþanan oylama skandalý karþýlýklý suçlamaya dönüþtü

Potada , Avrupa Þampiyonasý Eleme Grubu'nda Bulgaristan'a karþý alýnan farklý yenilgi tepkileri de beraberinde getirdi

== Meclis'te ==
☑ Meclis+Noun+Prop+A3sg+Pnon+Loc

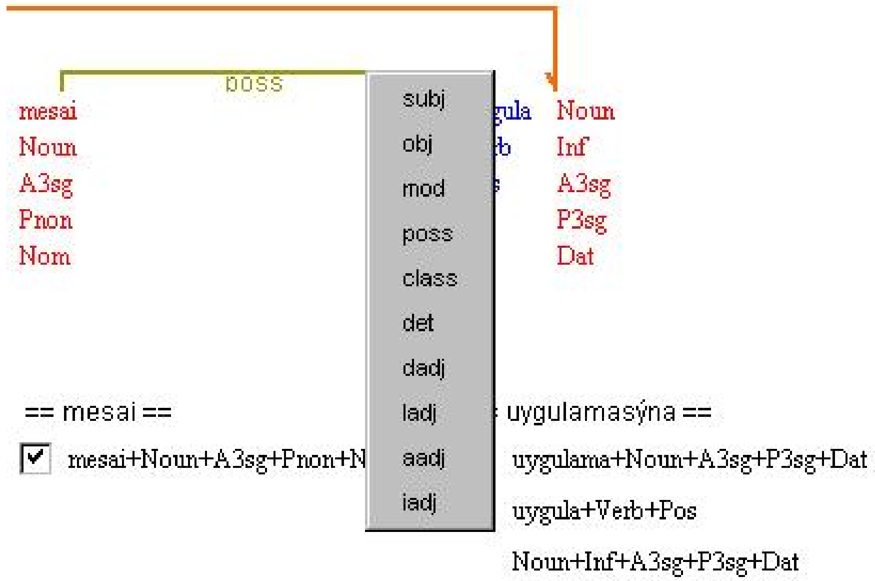
== yaþanan ==
☑ yaþa+Verb
Verb+Pass+Pos
Adj+PresPart

== oylama ==
□ oyla+Verb+Pos
Noun+Inf+A3sg+Pnon+Nom
☑ oyla+Verb+Neg+Imp+A2sg

== skandalý ==
□ skandal+Noun+A3sg+Pnon+Acc
☑ skandal+Noun+A3sg+P3sg+Nom

== karþýlýklý ==
□ karþýlýk+Noun+A3sg+Pn
Adj+With
□ karþýlýk+Noun+A3sg+Pn
Adj+With
Noun+Zero+A3sg+Pnon+N
☑ karþý+Noun+A3sg+Pnon+
Noun+Ness+A3sg+Pnon+N

Figure 1.6 Selecting the link type

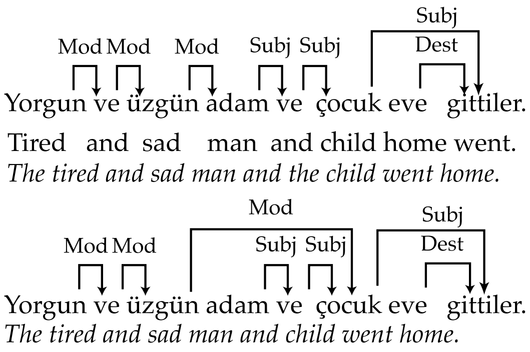


annotator to select a link type as shown in Figure 1.6. In a future version, this linking will be done in a more intelligent fashion with the destination IG and the contents of the label popup menu being determined by the source IG.

4. SOME DIFFICULT ISSUES

Turkish is a pro-drop language, and subject (and usually various other constituents) may be elided on the surface. In the case of subjects, the information is recoverable from the agreement marker on the verbs. Since we aim to capture just the surface relations, such covert cases are not marked. The cases for verb ellipsis is a bit more tricky. In these cases we have constituents which do not have a surface governor. We have for the time being opted to capture these cases by explicitly entering a dummy constituent (with a null surface form but nevertheless being a token) linked with a special link to the parallel verb, indicating its ellipsis status. Then the contituents of the elided verb can be attached to this dummy constituent.

Figure 1.7 Linking conjoined constituents


Mod Mod Mod Subj Subj Subj Dest
Yorgun ve üzgün adam ve çocuk eve gittiler.
Tired and sad man and child home went.
The tired and sad man and the child went home.
Mod Mod Mod Subj Subj Subj Dest
Yorgun ve üzgün adam ve çocuk eve gittiler.
The tired and sad man and child went home.

Headless constructions such as coordinating conjunctions have been one of the weaker points of dependency grammar approaches. Our solution for describing coordinate conjunction constructs essentially follows Järvinen and Tapanainen [9]. For a sequence of IGs like

$$D_1 \ldots C \ldots D_2 \ldots C \ldots\ldots D_k \ldots H$$

where D_i are the dependent IGs that are coordinated and Cs are the conjunction IGs (for , (comma), *and* and *or*), and H is the head IG, we effectively thread a "long link" from D_1 to H. If the link between D_k and H is labeled with l, then dependent D_i links to the following C with link l, and this C links to D_{i+1} with l. One feature of Turkish simplifies this threading a bit: the left conjunct IG has to immediately precede the conjunction IG (except for the very unlikely cases of verbal coordination in inverted constituent orders). Figure 1.7 shows the links for encoding two possible interpretations of conjunction scopes for a simple Turkish sentence.

5. CONCLUSIONS AND FUTURE WORK

As we mentioned at the outset, our current work has concentrated on resolving the issues in encoding Turkish treebanks. There are certainly other theoretical issues especially in the dependency representations of various problematic constructs. We have also completed the implementation a first version of an annotation tool for compiling Turkish treebanks.

Our current work involves developing and refining a set of guidelines for annotation Turkish text using this framework and developing a the final specifications of the second version of the annotation tool based on the experience gained from building and experimenting with the current tool. The new tool will integrate tools that we have already developed for tokenization, morphological analysis, collocation processing and morphological disambiguation [14, 18] as plug-ins in an extensible way. We also expect to utilize our statistical disambiguator module [6] and integrate a dependency parser (e.g., [15]) to generate full or partial dependency parses and have a human operator disambiguate and correct the parses if necessary.

Acknowledgments

This work is supported by a grant from TÜBİTAK, the Turkish Scientific and Technical Research Council by Grant No: EEEAG-199E026.

Appendix: Turkish Morphological Features

In this section we provide a list of morphological features used in the encoding of about 9,000 possible IGs that can be produced by our morphological analysis. Although not all of these have been used in examples used in this chapter, we feel it is useful for conveying to the reader the wealth of the information Turkish lexical forms encode.

- **Major Parts of Speech**: `+Noun, +Adj, +Adv, +Conj, +Det, +Dup, +Interj, +Ques, +Verb, +Postp, +Num, +Pron, +Punc`. (`+Dup` category contains onomatopoeia words which only appear as duplications in a sentence.)

- **Minor Parts of Speech**: These typically follow a major POS to further subdivide that class, or to indicate the kind of derivation involved.

  - After `+Num`: `+Card, +Ord, +Percent, +Range, +Real, +Ratio, +Distrib, +Time`.
  - After `+Noun`: `+Inf, +PastPart, FutPart, +Prop, +Zero`.
  - After `+Adj`: `+PastPart, +FutPart, +PresPart`.
  - After `+Pron`: `+DemonsP, +QuesP, +ReflexP, +PersP, +QuantP`.

- The following (mostly semantic) markers are used after derivations to indicate the kind of derivation involved:

  - After `+Adv` derived from verbs: `+AfterDoingSo, +SinceDoingSo, +As (he does it), +When, +ByDoingSo, +While, +AsIf, +WithoutHavingDoneSo`.
  - After `+Adv` derived from Adjectives: `+Ly` (equivalent to the English *+ly* derivation.)
  - After `+Adv` derived from temporal nouns: `+Since`
  - After `+Adj` derived from nouns: `+With, +Without +SuitableFor, +InBetween, +Rel`.
  - After `+Noun` derived from adjectives: `+Ness` (as in red vs. redness)
  - After `+Noun` derived from nouns: `+Agt` (someone involved in some way with the stem noun), `+Dim` (Diminutive),
  - After `+Verb` derived from nouns or adjectives: `+Become` (to become like the noun or adjective in the stem) `+Acquire` (to acquire the noun in the stem)
  - A `+Zero` appears after a zero morpheme derivation.

- Nominal forms (Nouns, Derived Nouns, Pronouns, Participles and Infinitives) get the following additional inflectional markers:

  1. **Number/Person Agreement**: `+A1sg, +A2sg, +A3sg, +A1pl, +A2pl, +A3pl`.
  2. **Possessive Agreement**: `+P1sg, +P2sg, +P3sg, +P1pl, +P2pl, +P3pl, +Pnon` (no overt agreement).
  3. **Case**:`+Nom, +Acc, +Dat, +Abl, +Loc, +Gen, +Ins`.

- Adjectives (lexical or derived) do not take any inflection, except `+Adj+PastPart` and `+Adj+FutPart` will have a `+Pxxx` (possessive agreement as above) to mark verbal agreement. Any other inflection to adjectives implies type-raising to nouns and the inflection goes onto the noun after a 0-morpheme derivation.

- Verbs have two sets of markers which are treated as derivations:

  1. **Voice**: `+Pass, +Caus, +Reflex +Recip`, (A verb form may have multiple causative markers).
  2. **Compounding/Modality**: `+Able` (able to verb), `+Repeat` (verb repeatedly), `+Hastily` (verb hastily), `+EverSince` (have been verb-ing ever since), `+Almost` (almost verb-ed but did not), `+Stay` (stayed frozen while verb-ing), `+Start` (start verb-ing immediately)

- Verbs also get the following inflectional markers:

  1. **Polarity**: `+Pos, +Neg`
  2. **Tense-Aspect-Mood**: A finite verb may have 1 or 2 of `+Past` (past tense), `+Narr` (narrative past tense), `+Fut` (future tense), `+Aor` (Aorist, may indicate habitual, present, future, you name it), `+Pres` (present tense, for predicative nominals or adjectives), `+Desr` (desire/wish), `+Cond` (conditional), `+Neces` (Necessitative, must), `+Opt` (optative, let me/him/her verb), `+Imp` (imperative), `+Prog1` (Present continuous, process), `+Prog2` (Present continuous, state).
  3. Verbs also have number person agreement markers (see nominal forms earlier) and an optional copula marker.

Notes

[1] Literally, "(the thing existing) at the time we caused (something) to become strong". Obviously this is not a word that one would use everyday. Turkish words (excluding noninflecting frequent words such as conjunctions, clitics etc) found in typical text average about 10 letters in length.

2. Please refer to the comprehensive list of morphological features given in Appendix A for the semantics of some of the non-obvious symbols used here.

3. Though they may be separated by various clitics, in which case the collocation can not be recognized by simple local means.

4. This however does not mean that there no non-projective constructs in Turkish. There are a number of constructs, such as an adverbial modifying a verb, cutting in between a modifier and the head noun making up the subject NP. These, however, are very rare. Our representation does not have any restriction regarding projectivity and lets us represent the crossing links in such case.

5. Words in this context may also be a lexicalized or non-lexicalized collocations.

6. The input to the annotator is actually morphologically preprocessed with each token already having been analyzed in all its ambiguities. This same file could also be run through a morphological disambiguator module [7]. If this disambiguator makes any mistakes (and they do), our current tool does not let us correct an incorrectly disambiguated morphological analyses yet, so we have opted not to disambiguated for the time being.

References

[1] A. Abeillé, L. Clément, and A. Kinyon. Building a treebank for french. In A. Abeillé, editor, *Building and Exploiting Syntatically Annotated Corpora*, Text, Speech and Language Technology. Kluwer, 2001.

[2] A. Bémová, J. Hajič, B. H. J. Panenová, A. Böhmova, and E. Hajicova. The Prague Dependency Treebank. In A. Abeillé, editor, *Building and Exploiting Syntatically Annotated Corpora*, Text, Speech and Language Technology. Kluwer, 2001.

[3] T. Brants, W. Skut, and H. Uszkoreit. Syntactic annotation of a german newspaper corpus. In A. Abeillé, editor, *Building and Exploiting Syntatically Annotated Corpora*, Text, Speech and Language Technology. Kluwer, 2001.

[4] E. Erguvanlı. *The Function of Word order in Turkish.* PhD thesis, University of California, Los Angeles, 1979.

[5] J. Hajič. Building a syntactically annotated corpus: The Prague Dependency Treebank. In E. Hajicova, editor, *Issues in Valency and Meaning: Studies in Honour of Jarmila Panenova.* Karolinum – Charles University Press, Prague, April 1998.

[6] D. Z. Hakkani-Tür. *Statistical Language Modeling for Turkish.* PhD thesis, Bilkent University, Department of Computer Science, Ankara, Turkey, 2000.

[7] D. Z. Hakkani-Tür, K. Oflazer, and G. Tür. Statistical morphological disambiguation for agglutinative languages. In *Proceedings of COLING 2000.* ICCL, August 2000.

[8] J. Hankamer. Morphological parsing and the lexicon. In W. Marslen-Wilson, editor, *Lexical Representation and Process.* MIT Press, 1989.

[9] T. Järvinen and P. Tapanainen. Towards an implementable dependency grammar. In *Proceedings of COLING/ACL'98 Workshop on Processing Dependency-based Grammars*, pages 1–10, 1998.

[10] Y. Lepage, A. Shin-Ichi, A. Susumu, and I. Hitoshi. An annotated corpus in Japanese using Tesniere's structural syntax. In *Proceedings of COLING-ACL'98 Workshop on the Processing of Dependency-based Grammars*, 1998.

[11] D. Lin. A dependency-based method for evaluation broad-coverage parsers. In *Proceedings of IJCAI'95*, 1995.

[12] M. Marcus, B. Santorini, and M. A. Marcinkiewitz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 1993.

[13] M. Marcus and A. Taylor. The Penn Treebank. In A. Abeillé, editor, *Building and Exploiting Syntatically Annotated Corpora*, Text, Speech and Language Technology. Kluwer, 2001.

[14] K. Oflazer. Two-level description of Turkish morphology. *Literary and Linguistic Computing*, 9(2), 1994.

[15] K. Oflazer. Dependency parsing with an extended finite state approach. In *Proceedings of ACL'99, the 37th Annual Meeting of the Association for Computational Linguistics*, 1999.

[16] K. Oflazer and İ. Kuruöz. Tagging and morphological disambiguation of Turkish text. In *Proceedings of the* 4^{th} *Applied Natural Language Processing Conference*, pages 144–149. ACL, October 1994.

[17] K. Oflazer and G. Tür. Combining hand-crafted rules and unsupervised learning in constraint-based morphological disambiguation. In E. Brill and K. Church, editors, *Proceedings of the ACL-SIGDAT Conference on Empirical Methods in Natural Language Processing*, 1996.

[18] K. Oflazer and G. Tür. Morphological disambiguation by voting constraints. In *Proceedings of ACL'97/EACL'97, The 35th Annual Meeting of the Association for Computational Linguistics*, June 1997.

[19] W. Skut, B. Krenn, T. Brants, and H. Uszkoreit. An annotation scheme for free word order languages. In *Proceedings of Fifth Conference on Applied Natural Language Processing*, 1997.