Automatic Induction of a CCG Grammar for Turkish

Ruken Çakıcı
School of Informatics
Institute for Communicating and Collaborative Systems
University of Edinburgh
2 Buccleuch Place, Edinburgh EH8 9LW
United Kingdom
r.cakici@sms.ed.ac.uk

Abstract

This paper presents the results of automatically inducing a Combinatory Categorial Grammar (CCG) lexicon from a Turkish dependency treebank. The fact that Turkish is an agglutinating free word-order language presents a challenge for language theories. We explored possible ways to obtain a compact lexicon, consistent with CCG principles, from a treebank which is an order of magnitude smaller than Penn WSJ.

1 Introduction

Turkish is an agglutinating language, a single word can be a sentence with tense, modality, polarity, and voice. It has free word-order, subject to discourse restrictions. All these properties make it a challenge to language theories like CCG (Steedman (2000)).

Several studies have been made into building a CCG for Turkish (Bozşahin, 2002; Hoffman, 1995). Bozşahin builds a morphemic lexicon to model the phrasal scope of the morphemes which cannot be acquired with classical lexemic approach. He handles scrambling with type raising and composition. Hoffman proposes a generalisation of CCG (Multiset-CCG) for argument scrambling. She underspecifies the directionality, which results in an undesirable increase in the generative power of the grammar. However, Baldridge (2002) gives a more restrictive form of free order CCG. Both Hoffman and Baldridge ignore morphology and treat the inflected forms as different words.

The rest of this section contains an overview of the underlying formalism (1.1). This is followed by a review of the relevant work (1.2). In Section 2, the properties of the data are explained. Section 3 then gives a brief sketch of the algorithm used to induce a CCG lexicon, with some examples of how certain phenomena in Turkish are handled. As is likely to be the case for most languages for the foreseeable future, the Turkish treebank is quite small (less than 60K words). A major emphasis in the project is on generalising the induced lexicon to improve coverage. Results and future work are discussed in the last two sections.

1.1 Combinatory Categorial Grammar

Combinatory Categorial Grammar (Ades and Steedman, 1982; Steedman, 2000) is an extension to the classical Categorial Grammar (CG) of Ajdukiewicz (1935) and Bar-Hillel (1953). CG, and extensions to it, are lexicalist approaches which deny the need for movement or deletion rules in syntax. Transparent composition of syntactic structures and semantic interpretations, and flexible constituency make CCG a preferred formalism for long-range dependencies and non-constituent coordination in many languages e.g. English, Turkish, Japanese, Irish, Dutch, Tagalog (Steedman, 2000; Baldridge, 2002).

The categories in categorial grammars can be atomic, or functions which specify the directionality of their arguments. A lexical item in a CG can be represented as the triplet: $\phi := \sigma : \lambda$ where ϕ is the phonological form, σ is its syntactic type, and λ its semantic type. Some examples are:

Proceedings of the ACL Student Research Workshop, pages 73–78,
Ann Arbor, Michigan, June 2005.

(1) a. $book := N: \mathsf{book}$
b. $oku := (S\backslash NP)\backslash NP: \lambda x.\lambda y.\mathsf{read}\ xy$

In classical CG, there are two kinds of application rules, which are presented below:

(2) Forward Application (>):
$X/Y: f \quad Y: a \quad \Rightarrow \quad X: fa$
Backward Application (<):
$Y: a \quad X\backslash Y: f \quad \Rightarrow \quad X: fa$

In addition to functional application rules, CCG has combinatory operators for composition (**B**), type raising (**T**), and substitution (**S**).[1] These operators increase the expressiveness to mildly context-sensitive while preserving the transparency of syntax and semantics during derivations, in contrast to the classical CG, which is context-free (Bar-Hillel et al., 1964).

(3) Forward Composition (>**B**):
$X/Y: f \quad Y/Z: g \quad \Rightarrow \quad X/Z: \lambda x.f(gx)$
Backward Composition (<**B**):
$Y\backslash Z: g \quad X\backslash Y: f \quad \Rightarrow \quad X\backslash Z: \lambda x.f(gx)$

(4) Forward Type Raising (>**T**):
$X: a \quad \Rightarrow \quad T/(T\backslash X): \lambda f.f[a]$
Backward Type Raising (<**T**):
$X: a \quad \Rightarrow \quad T\backslash(T/X): \lambda f.f[a]$

Composition and type raising are used to handle syntactic coordination and extraction in languages by providing a means to construct constituents that are not accepted as constituents in other theories.

1.2 Relevant Work

Julia Hockenmaier's robust CCG parser builds a CCG lexicon for English that is then used by a statistical model using the Penn Treebank as data (Hockenmaier, 2003). She extracts the lexical categories by translating the treebank trees to CCG derivation trees. As a result, the leaf nodes have CCG categories of the lexical entities. Head-complement distinction is not transparent in the Penn Treebank so Hockenmaier uses an algorithm to find the heads (Collins, 1999). There are some inherent advantages to our use of a dependency treebank that only represents surface dependencies. For example, the head is always known, because dependency links are from dependant to head. However, some problems are caused by that fact that only surface dependencies are included. These are discussed in Section 3.5.

[1] Substitution and others will not be mentioned here. Interested reader should refer to Steedman (2000).

2 Data

The METU-Sabancı Treebank is a subcorpus of the METU Turkish Corpus (Atalay et al., 2003; Oflazer et al., 2003). The samples in the corpus are taken from 3 daily newspapers, 87 journal issues and 201 books. The treebank has 5635 sentences.There are a total of 53993 tokens. The average sentence length is about 8 words. However, a Turkish word may correspond to several English words, since the morphological information which exists in the treebank represents additional information including part-of-speech, modality, tense, person, case, etc. The list of the syntactic relations used to model the dependency relations are the following.

1.Subject	2. Object	3.Modifier
4.Possessor	5.Classifier	6.Determiner
7.Adjunct	8.Coordination	9.Relativiser
10.Particles	11.S.Modifier	12.Intensifier
13. Vocative	14. Collocation	15. Sentence
16.ETOL		

ETOL is used for constructions very similar to phrasal verbs in English. "Collocation" is used for the idiomatic usages and word sequences with certain patterns. Punctuation marks do not play a role in the dependency structure unless they participate in a relation, such as the use of comma in coordination. The label "Sentence" links the head of the sentence to the punctuation mark or a conjunct in case of coordination. So the head of the sentence is always known, which is helpful in case of scrambling. Figure 1 shows how (5) is represented in the treebank.

(5) Kapının kenarındaki duvara dayanıp bize baktı bir an.
(He) looked at us leaning on the wall next to the door, for a moment.

The dependencies in Turkish treebank are surface dependencies. Phenomena such as traces and pro-drop are not modelled in the treebank. A word

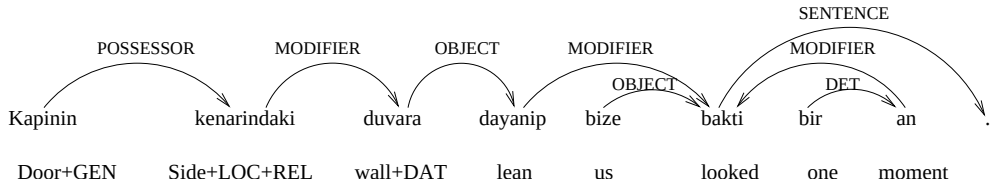

Figure 1: The graphical representation of the dependencies

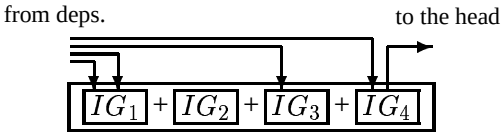

Figure 2: The structure of a word

can be dependent on only one word but words can have more than one dependants. The fact that the dependencies are from the head of one constituent to the head of another (Figure 2) makes it easier to recover the constituency information, compared to some other treebanks e.g. the Penn Treebank where no clue is given regarding the head of the constituents.

Two principles of CCG, Head Categorial Uniqueness and Lexical Head Government, mean both extracted and in situ arguments depend on the same category. This means that long-range dependencies must be recovered and added to the trees to be used in the lexicon induction process to avoid wrong predicate argument structures (Section 3.5).

3 Algorithm

The lexicon induction procedure is recursive on the arguments of the head of the main clause. It is called for every sentence and gives a list of the words with categories. This procedure is called in a loop to account for all sentential conjuncts in case of coordination (Figure 3).

Long-range dependencies, which are crucial for natural language understanding, are not modelled in the Turkish data. Hockenmaier handles them by making use of traces in the Penn Treebank (Hockenmaier, 2003)[sec 3.9]. Since Turkish data do not have traces, this information needs to be recovered from morphological and syntactic clues. There are no relative pronouns in Turkish. Subject and object extraction, control and many other phenomena are marked by morphological processes on the subordinate verb. However, the relative morphemes behave in a similar manner to relative pronouns in English (Çakıcı, 2002). This provides the basis for a heuristic method for recovering long range dependencies in extractions of this type, described in Section 3.5.

```
recursiveFunction(index i, Sentence s)
headcat = findheadscat(i)
//base case
if myrel is “MODIFIER”
        handleMod(headcat)
elseif “COORDINATION”
        handleCoor(headcat)
elseif “OBJECT”
        cat = NP
elseif “SUBJECT”
        cat = NP[nom]
elseif “SENTENCE”
        cat = S
.
.
if hasObject(i)
        combCat(cat,“NP”)
if hasSubject(i)
        combCat(cat,“NP[nom]”)
//recursive case
        forall arguments in arglist
                recursiveFunction(argument,s);
```

Figure 3: The lexicon induction algorithm

3.1 Pro-drop

The subject of a sentence and the genitive pronoun in possessive constructions can drop if there are morphological cues on the verb or the possessee. There is no pro-drop information in the treebank, which is consistent with the surface dependency

approach. A *[nom]* (for nominative case) feature is added to the NPs by us to remove the ambiguity for verb categories. All sentences must have a nominative subject.[2] Thus, a verb with a category *S\NP* is assumed to be transitive. This information will be useful in generalising the lexicon during future work (Section 5).

	original	pro-drop
transitive	(S\NP[nom])\NP	S\NP
intransitive	S\NP[nom]	S

3.2 Adjuncts

Adjuncts can be given CCG categories like *S/S* when they modify sentence heads. However, adjuncts can modify other adjuncts, too. In this case we may end up with categories like (6), and even more complex ones. CCG's composition rule (3) means that as long as adjuncts are adjacent they can all have *S/S* categories, and they will compose to a single *S/S* at the end without compromising the semantics. This method eliminates many gigantic adjunct categories with sparse counts from the lexicon, following (Hockenmaier, 2003).

(6) daha:= *(((S/S)/(S/S))/((S/S)/(S/S)))/*
(((S/S)/(S/S))/((S/S)/(S/S)))
'more'

3.3 Coordination

The treebank annotation for a typical coordination example is shown in (7). The constituent which is directly dependent on the head of the sentence, "zıplayarak" in this case, takes its category according to the algorithm. Then, conjunctive operator is given the category *(X\X)/X* where *X* is the category of "zıplayarak" (or whatever the category of the last conjunct is), and the first conjunct takes the same category as *X*. The information in the treebank is not enough to distinguish sentential coordination and VP coordination. There are about 800 sentences of this type. We decided to leave them out to be annotated appropriately in the future.

(7)

	Mod.	Coor.	Mod.	Sentence	
Koşarak	ve	zıplayarak	geldi	.	

He came running and jumping.

[2]This includes the passive sentences in the treebank

3.4 NPs

Object heads are given *NP* categories. Subject heads are given *NP[nom]*. The category for a modifier of a subject NP is *NP[nom]/NP[nom]* and the modifier for an object NP is *NP/NP* since NPs are almost always head-final.

3.5 Subordination and Relativisation

The treebank does not have traces or null elements. There is no explicit evidence of extraction in the treebank; for example, the heads of the relative clauses are represented as modifiers. In order to have the same category type for all occurences of a verb to satisfy the Principle of Head Categorial Uniqueness, heuristics to detect subordination and extraction play an important role.

(8) Kitabı okuyan adam uyudu.
Book+ACC read+PRESPART man slept.
The man who read the book slept

These heuristics consist of morphological information like existence of a "PRESPART" morpheme in (8), and part-of-speech of the word. However, there is still a problem in cases like (9a) and (9b). Since case information is lost in Turkish extractions, surface dependencies are not enough to differentiate between an adjunct extraction (9a) and an object extraction (9b). A *T.LOCATIVE.ADJUNCT* dependency link is added from "araba" to "uyuduğum" to emphasize that the predicate is intransitive and it may have a locative adjunct. Similarly, a *T.OBJECT* link is added from "kitap" to "okuduğum". Similar labels were added to the treebank manually for approximately 800 sentences.

(9) a. Uyuduğum araba yandı.
Sleep+PASTPART car burn+PAST.
The car I slept in burned.

b. Okuduğum kitap yandı.
Read+PASTPART book burn+PAST.
The book I read burned.

The relativised verb in (9b) is given a transitive verb category with pro-drop, *(S\NP)*, instead of *(NP/NP)\NP*, as the Principle of Head Categorial Uniqueness requires. However, to complete the process we need the relative pronoun equivalent in Turkish,*-dHk+AGR*. A lexical entry with

category *(NP/NP)\(S\NP)* is created and added to the lexicon to give the categories in (10) following Bozşahin (2002).[3]

(10)

Oku	-duğum	kitap	yandı.
S\NP	(NP/NP)\(S\NP)	NP	S\NP

4 Results

The output is a file with all the words and their CCG categories. The frequency information is also included so that it can be used in probabilistic parsing.

The most frequent words and their most frequent categories are given in Figure 4. The fact that the 8th most frequent word is the non-function word "dedi"(*said*) reveals the nature of the sources of the data —mostly newspapers and novels.

In Figure 5 the most frequent category types are shown. The distribution reflects the real usage of the language (some interesting categories are explained in the last column of the table). There are 518 distinct category types in total at the moment and 198 of them occur only once, but this is due to the fact that the treebank is relatively small (and there are quite a number of annotation mistakes in the version we are using).

In comparison with the English treebank lexicon (1224 types with around 417 occuring only once (Hockenmaier, 2003)) this probably is not a complete inventory of category types. It may be that dependency relations are too few to make the correct category assignment automatically. For instance, all adjectives and adverbs are marked as "MODIFIER". Figure 6 shows that even after 4500 sentences the curve for most frequent categories has not converged. The data set is too small to give convergence and category types are still being added as unseen words appear. Hockenmaier (2003) shows that the curve for categories with frequencies greater than 5 starts to converge only after 10K sentences in the Penn Treebank.[4]

[3]Current version of the treebank has empty "MORPH" fields. Therefore, we are using dummy tokens for relative morphemes at the moment.

[4]The slight increase after 3800 sentences may be because the data are not uniform. Relatively longer sentences from a history article start after short sentences from a novel.

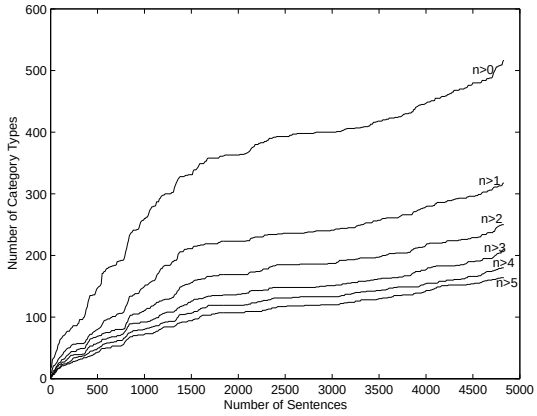

Figure 6: The growth of category types

5 Future Work

The lexicon is going to be trained and tested with a version of the statistical parser written by Hockenmaier (2003). There may be some alterations to the parser, since we will have to use different features to the ones that she used, such as morphological information.

Since the treebank is considerably small compared to the Penn WSJ treebank, generalisation of the lexicon and smoothing techniques will play a crucial role. Considering that there are many small-scale treebanks being developed for "understudied" languages, it is important to explore ways to boost the performances of statistical parsers from small amounts of human labeled data.

Generalisation of this lexicon using the formalism in Baldridge (2002) would result in a more compact lexicon, since a single entry would be enough for several word order permutations. We also expect that the more effective use of morphological information will give better results in terms of parsing performance. We are also considering the use of unlabelled data to learn word-category pairs.

References

A.E. Ades and Mark Steedman. 1982. On the order of words. *Linguistics and Philosophy*, 4:517–558.

Kazimierz Ajdukiewicz. 1935. Die syntaktische konnexitat. In *Polish Logic, ed. Storrs McCall, Oxford University Press*, pages 207–231.

token	eng.	freq.	pos	most freq. cat	fwc*
,	Comma	2286	Conj	(NP/NP)\NP	159
bir	a	816	Det	NP/NP	373
-yAn	who	554	Rel. morph.	(NP/NP)\(S\NP)	554
ve	and	372	Conj	(NP/NP)\NP	100
de	too	335	Int	NP[nom]\NP[nom]	116
bu	this	279	Det	NP/NP	110
da	too	268	Int	NP[nom]\NP[nom]	86
dedi	said	188	Verb	S\NP	87
-DHk+AGR	which	163	Rel. morph.	(NP/NP)\(S\NP)	163
Bu	This	159	Det	NP/NP	38
gibi	like	148	Postp	(S/S)\NP	21
o	that	141	Det	NP/NP	37

*__fwc__ Frequency of the word occuring with the given category

Figure 4: The lexicon statistics

cattype	frequency	rank	type
NP	5384	1	noun phrase
NP/NP	3292	2	adjective,determiner, etc
NP[nom]	3264	3	subject NP
S/S	3212	4	sentential adjunct
S\NP	1883	5	transitive verb with pro-drop
S	1346	6	sentence
S\NP[nom]	1320	7	intransitive verb
(S\NP[nom])\NP	827	9	transitive verb

Figure 5: The most frequent category types

Nart B. Atalay, Kemal Oflazer, and Bilge Say. 2003. The annotation process in the Turkish Treebank. In *Proceedings of the EACL Workshop on Linguistically Interpreted Corpora*, Budapest, Hungary.

Jason M. Baldridge. 2002. *Lexically Specified Derivation Control in Combinatory Categorial Grammar*. Ph.D. thesis, University of Edinburgh.

Yehoshua Bar-Hillel, C. Gaifman, and E. Shamir. 1964. On categorial and phrase structure grammars. In *Language and Information ed. Bar-Hillel, Addison-Wesley*, pages 99–115.

Yehoshua Bar-Hillel. 1953. A quasi-arithmetic description for syntactic description. *Language*, 29:47–58.

Cem Bozşahin. 2002. The combinatory morphemic lexicon. *Computational Linguistics*, 28(2):145–186.

Ruken Çakıcı. 2002. A computational interface for syntax and morphemic lexicons. Master's thesis, Middle East Technical University.

Michael Collins. 1999. *Head-driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Julia Hockenmaier. 2003. *Data Models for statistical parsing with Combinatory Categorial Grammar*. Ph.D. thesis, University of Edinburgh.

Beryl Hoffman. 1995. *The Computational Analysis of the Syntax and Interpretation of "Free" Word Order in Turkish*. Ph.D. thesis, University of Pennsylvania.

Kemal Oflazer, Bilge Say, Dilek Zeynep Hakkani-Tür, and Gokhan Tür. 2003. Building a turkish treebank. In Abeille Anne, editor, *Treebanks: Building and Using Parsed Corpora*, pages 261–277. Kluwer, Dordrecht.

Mark Steedman. 2000. *The Syntactic Process*. The MIT Press, Cambridge, Massachusetts.