CS674/INFO630: Advanced Language Technologies, Fall 2007, Lecture by Lillian Lee
Lecture 21 Guide: Augmentations to Context-Free Grammars

Alex Chao David Collins

November 13, 2007

1 Introduction

Today we will be discussing various augmented versions of context-free-grammars (CFGs). This material is based upon James Allen's 1995 textbook [1].

One may wonder why anyone would even need to study sentence structure, as the bag-of-words models we have been discussing seem to work fine for information retrieval (IR) applications. As it turns out, many applications of analyzing sentential structure are not in IR but rather in areas such as information extraction, document summarization (or sentence compression), machine translation, and language generation. (This, however, is not to say that models of sentential structure cannot also help IR.) We will be discussing each of the above applications in some detail before we begin discussing augmented context-free-grammars.

2 Applications of Representing Sentence Structure

2.1 Information Extraction

One main application of the analysis of sentential structure of text is information extraction, which allows one to pull facts or assertions out of documents. For example, one can imagine using a collection of documents to populate a database with facts. A system with the purpose of answering questions like "how tall is Mt. Everest" might query this database. (As a side note, one might wonder whether Google hard codes the answers to some queries like this in its search results, although it clearly does not do so for all such queries. But the point is that the fact that Google does this indicates that they believe question answering to be a useful application that people wish to use.) To build a system like this, one would need to understand the syntactic structure of a sentence, rather than simply the collection of words in each document. However, consider the following sentence:

Joe saw Mary's mother with the sniper scope.

The syntactic structure of the sentence is correct, but is has two semantic interpretations; either Joe or Mary's mother could have the sniper scope. The point is that one must determine the person possessing the sniper scope before this information can be extracted.

2.2 Document Summarization/Sentence Compression

Now suppose we want to summarize, or compress, the following sentence so only the important parts of the sentence remain:

Betta put her signature ginger cookies on the table with a flourish.

We could compress the sentence as

Betta put her cookies on the table.

However, we could not compress the sentence as

Betta put her cookies.

Or as

Betta put her cookies with a flourish.

In order to distinguish between proper and improper compressions of the sentence, we need some way of knowing which parts of the sentence are required and which parts are optional.

2.3 Machine Translation

Another popular application of the analysis of sentential structure is machine translation. The sentential structure differs significantly among languages. In English, for instance, the syntactic structure of a sentence usually consists of a subject, then a verb, then an object (SVO). Compare that sentential structure with the structures of Hindi and Irish, which usually obey the orders SOV and VSO, respectively. Russian, in contrast to all of the above languages, has no particular order of the subject, verb, and object, but the roles of these structures in the sentence are usually marked. As another example of difference in sentential structure between languages, in English, the noun usually follows the adjective, while in French the opposite is usually true. You need syntactic structure on both sides (the source and the target languages) in order to properly represent the syntactic structure of any language.

2.4 Language Generation

Language generation can involve creating language output not only from other language data as in the summarization and machine translation applications considered above, but also from non-textual data. One can imagine various applications of this, including generating weather reports from meteorological data, or summaries of sporting events from a database-style event log. Typically, the output generated must have some understandable grammatical structure.

3 Modeling Sentential Structure

We will now begin our discussion of how to model sentential structure. Recall that CFGs give one level of decomposition at a time. There are two main issues with a basic CFG scheme:

- CFGs describe legal 1-step decompositions, but there may be many different types of constituents, causing there to be an unmanageable number of constituent labels and decompositions.
- Language has long-distance dependencies between constituents that CFGs cannot obviously manage. The direct object of a sentence, for instance, could be in one of multiple parts of a sentence, but it cannot be in both simultaneously. The direct objects relationship with the verb is difficult to enforce with basic CFGs. Moreover, to handle long-distance relationships, some formalisms put heads into more general constituent labels, thereby building a lexicalized CFG; but this dramatically scales up the number of possible constituent labels.

In a recent lecture we saw the *wh*-movement phenomenon. Consider the following example:

$$\text{Police } [[\text{informed}]_V [\text{me}]_{NP}]_{VP}.$$

$$VP \rightarrow V \ NP$$

The head *me* is the direct object of the sentence, and there are certain restrictions on any word that is substituted in its place. For example, any direct object that is substituted for *me* must have the property of being animate in order for the sentence to make sense.

Note that we could rearrange the above sentence into the following form:

$$[\text{Whom}]_{NP} \text{ did police } [[\text{inform}]_V]_{VP} \ ?$$

$$VP \rightarrow V$$

Notice that the verb phrase VP has a second decomposition rule, but neither this rule or the aforementioned rule account for the apparent "movement" of the direct object. We would really only like to have one expansion rule for this verb (namely, $VP \rightarrow V \ NP$), which should also handle *wh*-movement or other similar scenarios.

4 Traces

One proposed approach to dealing with movement phenomena involves the use of what have been coined *traces*. We can define the trace (donated by ε) as a "phonologically null" string (essentially a silent and empty word) that attempts to account for an item that has moved in the sentence structure. In a CFG, we can account for the trace with the following rule:

$$NP \rightarrow \varepsilon$$

In the case of the aforementioned sentence, *Whom* takes the on the role of a noun phrase that has moved from its expected position after *inform*. In the following construction, ε is the "trace" of the noun phrase *Whom*, which moved to the front of the sentence.

$$[\text{Whom}]_{NP} \text{ did the police } [\text{inform } [\varepsilon]_{NP}]_{VP} \text{ ?}$$

Notice that thus, the single rule $VP \rightarrow V\ NP$ suffices. In such a case, $[\varepsilon]_{NP}$ is called the *gap*, while $[\text{Whom}]_{NP}$ is called the *filler*. The filler and the gap correspond to each other or, equivalently, are said to *coordinate*. However, nothing in this informal construction necessarily enforces this coordination, so we should thus wonder how it might be handled by a CFG.

5 Feature-based CFGs

Feature-based CFGs (FBCFGs) constitute a solution to the stated problems that focuses on the lexicon and ideas we would like to represent with our grammar[1]. They are, in this sense, an "engineering solution". Through added notation and mechanisms, FBCFGs encode features of the lexicon in feature structures, which take the form of recursive attribute-value lists. These lists are recursive in that the values can themselves be feature lists. The lexical entry for *informed* is provided as a simple example.

$$\begin{bmatrix} CAT: & V \\ VFORM: & past \\ ROOT: & inform \end{bmatrix}$$

This simple entry contains the CATegory (V for *verb*), $VFORM$ (verb form), and $ROOT$ form of *informed*. It is noteworthy that the $ROOT$ behaves as a pointer, specifying inheritance of the features of the "base" form of the word. The entry for *inform* provides a more complex example.

$$\begin{bmatrix} CAT: & V & & \\ VFORM: & base & & \\ ROOT: & inform & & \\ SUBCAT: & \begin{bmatrix} 1: & \begin{bmatrix} CAT: & NP \\ RES: & animate \\ CASE: & \{DO,\ \_\_\} \end{bmatrix} \end{bmatrix} & & \end{bmatrix}$$

Note that this entry appropriately denotes *inform* as the "base" form of the verb. The $SUBCAT$egorization attribute defines a further set of feature lists for the arguments of the word in question. In this case, the $SUBCAT$egorization is that of a noun phrase, with the restriction (RES) that the noun phrase must refer to something animate. The $CASE$ attribute asserts that *inform* takes the noun phrase in question as a direct object (DO)[2]. The underscore in the $CASE$ set allows for nouns where case is not indicated, such as the proper noun "Bob" or the common noun "president".

We now return to the rule for the verb phrase, ($VP \rightarrow V\ NP$). In particular, we would like any restrictions on the argument to be enforced if movement occurs, and so would like some mechanism to allow these restrictions to be passed around the tree.

[1] In what follows, our focus is on the general ideas rather than on specific details of how these ideas should be implemented.

[2] DO is actually a slight abuse of notation. More formally, the $CASE$ should be specified as ACC or "accusative".

$$\begin{bmatrix} CAT: & VP \\ GAPINFO: & ?g \end{bmatrix} \rightarrow \begin{bmatrix} CAT: & V \\ SUBCAT: & [\ 1: \quad ?a\ ] \end{bmatrix} \begin{bmatrix} ?a \\ GAPINFO: & ?g \end{bmatrix}$$

The correspondence between the variable $?a$ in the *SUBCAT* feature list and the variable $?a$ in the right-most term is an instance of *co-indexing* and captures a unification constraint. That is, the second constituent in the decomposition must conform to the constraints specified by the *SUBCAT* feature list of the verb entry. This dependency ensures that no inconsistencies will arise between instantiations of terms. Similarly, variable $?g$ accounts for cases where the verb phrase involves a gap. *GAPINFO* essentially gets passed from the last constituent up to the verb phrase (left side of the rule) via variable $?g$. From there, it may further propagate its way up to the sentence and back down to the intended filler noun phrase.

Given the likelihood of such gap-filler dependencies, we would also like to have a rule to generate a trace. Such a rule might look like the following.

$$\begin{bmatrix} CAT: & ?c \\ CASE: & ?case \\ RES: & ?r \\ GAPINFO: & \begin{bmatrix} CAT: & ?c \\ CASE: & ?case \\ RES: & ?r \end{bmatrix} \\ NULL: & + \end{bmatrix} \rightarrow \varepsilon$$

Note that the gap effectively inherits its feature list from the feature list of the category that generated it. For completeness, we consider a rule required to generate the filler noun phrase in the sentence (this is, in fact, not the most or only linguistically plausible rule). Here, *GAPINFO* from the verb phrase *VP* is passed to the noun phrase *NP*, as a way of "injecting" the desired features. *NULL* indicates that gapping is disallowed in the *wh*-phrase.

$$\begin{bmatrix} CAT: & S \\ WH\text{-}MV: & + \end{bmatrix} \rightarrow \begin{bmatrix} CAT: & NP \\ NFORM: & pronoun \\ NULL: & - \end{bmatrix} \ did\ police \begin{bmatrix} CAT: & VP \\ GAPINFO: & ?g \\ VFORM: & base \end{bmatrix}$$

Ultimately, features provide an intuitive method for enforcing the often implicit constraints of natural languages, but as the discussion above should demonstrate, their implementation is nonetheless complicated and verbose. We are advised to take a step back and recall that if there were no long distance dependencies to begin with, we would not be forced to do deal with 'transmitting' features between distant but correlated entities. The next lecture explores whether such a formalism exists.

6 Sample Questions

Question 1

Suppose we live in a simple world of text in which there are only nouns, verbs, adjectives, and the word 'The' from the English language.

All legal sentences are decomposed as follows:

$$S \rightarrow \text{`The'}\ A\ N\ V,$$

where A is any adjective, N is a singular noun, and V is the third-person singular form of a verb. Suppose A, N, and V include all adjectives, singular nouns, and third-person singular verbs from the English language, respectively.

The following are example sentences that are valid.

The hungry dog begs. and *The wise prophet predicts.*

We will ignore the semantic meaning of the language such that every A-N-V configuration is valid.

a. Why is it not reasonable to ignore the semantic meaning of a sentence in such a way?

b. Instead of ignoring the the semantic meaning of the sentence in such a way, let us bring category proliferation into the picture. Suppose we constrain our sentence as follows:

   - The nouns can be divided into 4 subsets, call them N_1, N_2, N_3, N_4.
   - The verbs can be divided into 2 subsets, call them V_1 and V_2.
   - the adjectives can be divided into 2 subsets, call them A_1 and A_2.
   - N_1 can only appear in the same sentence with V_1 and A_1, or V_1 and A_2.
   - N_2 can only appear in the same sentence with V_2 and A_1.
   - N_3 can only appear in the same sentence with V_1 and A_1, or V_2 and A_1.
   - V_1 can appear in the same sentence with A_1, but not with A_2.

   Based on the above restrictions, find a smallest collection of possible decompositions of the form $S \rightarrow$ 'The' $A'\ N'\ V'$ that is exhaustive of the entire language but does not admit any sentences not in the language, where A', N', and V' are subsets of A, N, and V, respectively. For the purpose of the problem, you may use the "or" notation to specify decompositions. For example, one valid decomposition might be

   $S \rightarrow$ 'The' (A_1 or A_2) N_4 V_2.

c. How many decompositions are in the largest set of decompositions of S, where the subsets N_1, N_2, N_3, N_4, V_1, V_2, A_1, and A_2 are the smallest non-terminals that can be used in a decomposition?

Question 2

What was the main motivation for using feature-based CFGs? What benefits do they offer over conventional CFGs? Why is it seemingly important to have expansion rules for given nonterminals contain all required arguments, even if those arguments are phonologically null?

7 Sample Answers

Answer 1

a. Clearly, not all configurations of such nouns, adjectives and verbs make sense together. For example, the sentence

The vaporous dog flies.

would not make any semantic sense. In any reasonable and useful CFG scheme, there would be restrictions on the co-existence of verbs, adjectives, and nouns in a sentence that defines which configurations cause the sentence to make semantic sense.

b. The following collection of decompositions has the smallest number of decompositions possible to represent the entire language..

- $S \rightarrow$ 'The' A_1 (N_1 or N_3 or N_4) V_1
- $S \rightarrow$ 'The' A_1 (N_2 or N_3 or N_4) V_2
- $S \rightarrow$ 'The' A_2 N_4 V_2

c. The following seven decompositions are in the largest set of decompositions:

- $S \rightarrow$ 'The' A_1 N_1 V_1
- $S \rightarrow$ 'The' A_1 N_3 V_1
- $S \rightarrow$ 'The' A_1 N_4 V_1
- $S \rightarrow$ 'The' A_1 N_2 V_2
- $S \rightarrow$ 'The' A_1 N_3 V_2
- $S \rightarrow$ 'The' A_1 N_4 V_2
- $S \rightarrow$ 'The' A_2 N_4 V_2

Answer 2

Both FBCFGs and conventional CFGs focus on the properties of the lexicon and are able to account for many of the various parameters of natural languages, including tenses and cases. Additionally, they both allow one to encode semantic restrictions on lexical productions in order to prohibit awkward sentences like *The police informed the barricades.* However, in general, FBCFGs are more expressive than non-featured-based CFGs because they allow one to express abstract constraints through variables rather than forcing one to instantiate all legitimate values for the variables participating in the constraints. Achieving this expressiveness in a conventional CFG is much more difficult and inelegant.

To explain the importance of having expansion rules in which all arguments are represented, even if they are not 'phonologically present', consider the nonterminal VP. If we allowed for the expansion $VP \rightarrow V$ in addition to the expansion $VP \rightarrow V\ NP$, we would allow for verb phrases containing any verb and no object. While we would be able to produce sentences such as *Alice sings* or *John plays*, we would also have to allow for productions of seemingly meaningless sentences like *Kim puts.* Using a single and more general production rule, such as $VP \rightarrow V\ NP$, allows us to place constraints (using FBCFGs, for example) on the kinds of terminals that ultimately constitute the sentence.

References

[1] James Allen. Section 5.3: "Handling questions in context-free grammars". Natural Language Understanding, second edition. Benjamin/Cummings (1995).