

Inf 703 Information Organisation

Lecture Notes on Information Retrieval III: Natural Language Processing in Information Retrieval

Jagdish S. Gangolly

April 15, 1998

Some Perlisisms:

Syntactic sugar causes cancer of the semicolon.

It is better to have 100 functions operate on one data structure than 10 functions on 10 data structures.

Get into a rut early: Do the same process the same way. Accumulate idioms. Standardize. The only difference(!) between Shakespeare and you was the size of his idiom list - not the size of his vocabulary.

Recursion is the root of computation since it trades description for time.

Wherever there is modularity there is the potential for misunderstanding: Hiding information implies a need to check communication.

Perhaps if we wrote programs from childhood on, as adults we'd be able to read them.

Simplicity does not precede complexity, but follows it.

The string is a stark data structure and everywhere it is passed there is much duplication of process. It is a perfect vehicle for hiding information.

Everyone can be taught to sculpt: Michelangelo would have had to be taught not to. So it is with great programmers.

In software systems, it is often the early bird that makes the worm.

Giving up on assembly language was the apple in our Garden of Eden: Languages whose use squanders machine cycles are sinful. The LISP machine now permits LISP programmers to abandon bra and fig-leaf.

So many good ideas are never heard from again once they embark in a voyage on the semantic gulf.

Fools ignore complexity. Pragmatists suffer it. Some can avoid it. Geniuses remove it.

When we write programs that "learn", it turns out that we do and they don't.

Documentation is like term insurance: It satisfies because almost no one who subscribes to it depends on its benefits.

A year spent in artificial intelligence is enough to make one believe in God.

If your computer speaks English, it was probably made in Japan.

Prolonged contact with the computer turns mathematicians into clerks and vice versa.

We are on the verge: Today our program proved Fermat's next-to-last theorem.

What is the difference between a Turing machine and the modern computer? It's the same as that between Hillary's ascent of Everest and the establishment of a Hilton hotel on its peak.

The computer is the ultimate polluter: its feces are indistinguishable from the food it produces.

Whenever two programmers meet to criticize their programs, both are silent.

Why did the Roman Empire collapse? What is Latin for office automation?

It goes against the grain of modern education to teach children to program. What fun is there in making plans, acquiring discipline in organizing thoughts, devoting attention to detail and learning to be self-critical?

You think you know when you can learn, are more sure when you can write, even more when you can teach, but certain when you can program.

1 Introduction

In this handout, I shall provide the background needed to appreciate the papers assigned for this part of the course.

2 Text Analysis

- Morphological Decomposition (stemming – removal of affixes/suffixes)
- Concordances/KWIC indices

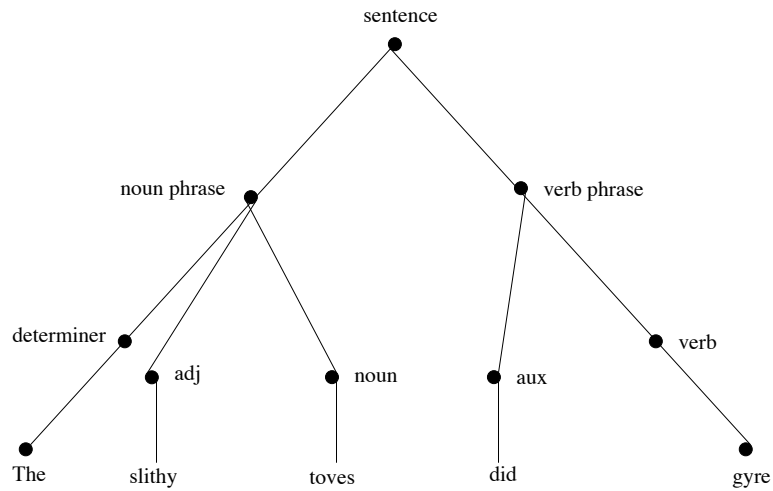


Figure 1: Parse Tree

2.1 Syntactic Analysis

2.2 Grammar & the Parsing Problem: Top-down/Bottom-up Parsing

The example from Robert Kowalski's *Logic for Problem Solving*. The parse tree is in Fig. 1.

Sentence: *The slithy toves did gyre.*

Grammar

- $$\begin{aligned}
 S &\rightarrow NP VP && (1) \\
 NP &\rightarrow Det Adj N && (2) \\
 VP &\rightarrow aux V && (3) \\
 Det &\rightarrow The && (4) \\
 adj &\rightarrow slithy && (5) \\
 N &\rightarrow toves && (6) \\
 aux &\rightarrow did && (7) \\
 V &\rightarrow gyre && (8)
 \end{aligned}$$

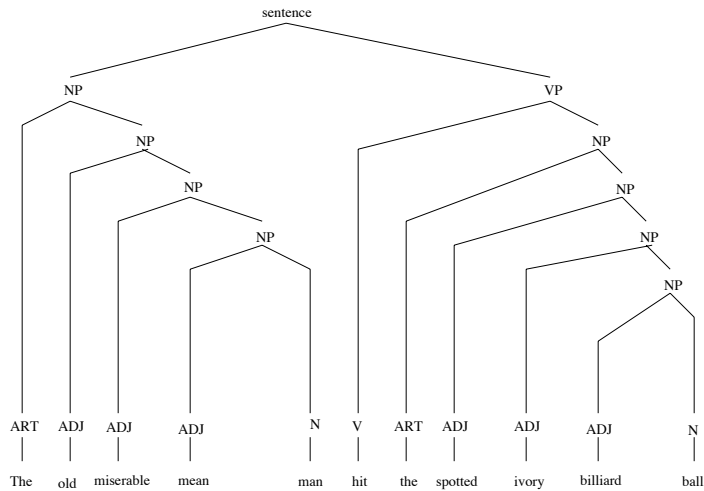


Figure 2: Parse Tree for the Second Example

2.3 Pattern matching Systems (ELIZA)

2.4 Grammar: Phrase structure, Context-Free, Augmented Transition Networks (ATN)

2.4.1 An Example:

Sentence: *The old miserable mean man hit the spotted ivory billiard ball.*

Grammar in [panini]-Backus-Naur (BNF) Notation

A. Rewrite rules:

$$\langle S \rangle ::= \langle NP \rangle \langle VP \rangle \quad (9)$$

$$\langle NP \rangle ::= \langle N \rangle \langle ADJ \rangle \langle NP \rangle \mid \langle ART \rangle \langle NP \rangle \quad (10)$$

$$\langle VP \rangle ::= \langle V \rangle \langle NP \rangle \quad (11)$$

B. Lexicon:

$$\langle ART \rangle ::= The \quad (12)$$

$$\langle N \rangle ::= man \mid ball \quad (13)$$

$$\langle ADJ \rangle ::= old \mid miserable \mid mean \mid spotted \mid ivory \mid billiard \quad (14)$$

The parse tree is in Figure 2, the ATNs are in Figure 3.

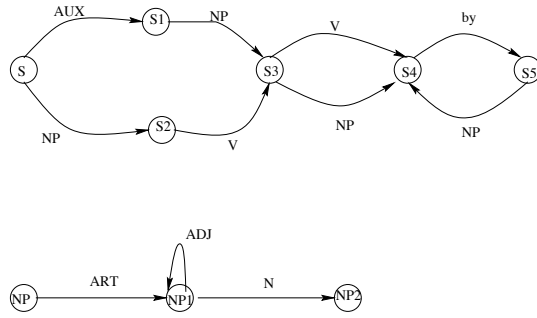


Figure 3: Augmented Transition Network

3 Conceptual Graphs

In a *conceptual graph*, there are two kinds of nodes, viz., *concepts* and *concept relations*. Every conceptual relation has one or more *arcs* which must be linked to some *concept*. Figure 4 gives a conceptual graph for an arch (From Sowa's *Conceptual Structures: Information Processing in Mind and Machine*).

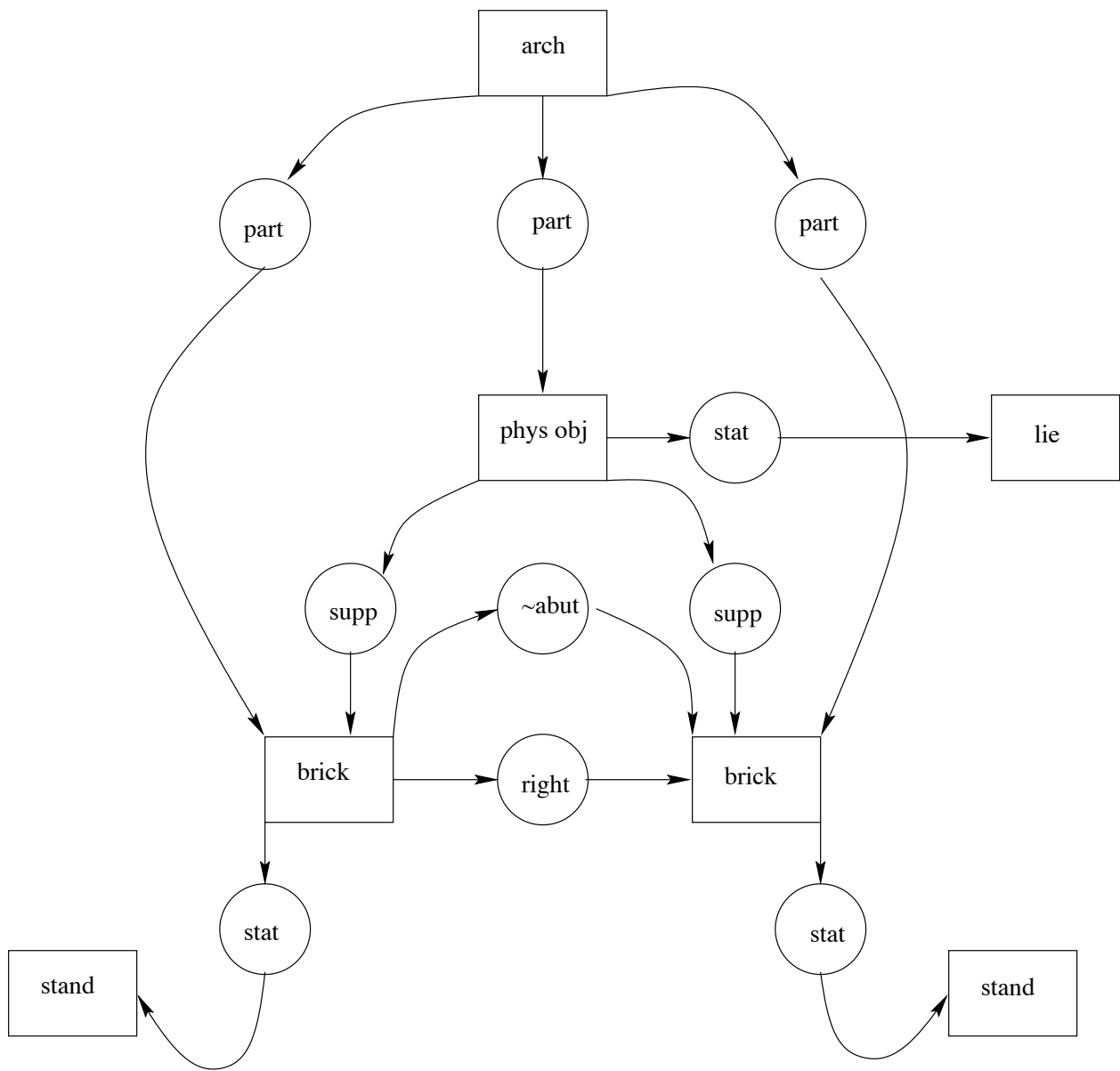


Figure 4: Conceptual graph for an arch