Special Topics in Computer Science

NLP in a Nutshell

CS492B Spring Semester 2009

Jong C. Park

Computer Science Department

Korea Advanced Institute of Science and Technology

PARTIAL PARSING

Partial Parsing

- Is Syntax Necessary?
- Word Spotting and Template Matching
  - ELIZA
  - Word Spotting in Prolog
- Multiword Detection
  - Multiwords
  - A Standard Multiword Annotation
  - Detecting Multiwords with Rules
  - The Longest Match
  - Running the Program

Partial Parsing

- Noun Groups and Verb Groups
  - Groups Versus Recursive Phrases
  - DCG Rules to Detect Noun Groups
  - DCG Rules to Detect Verb Groups
  - Running the Rules
- Group Detection as a Tagging Problem
  - Tagging Gaps
  - Tagging Words
  - Using Symbolic Rules
  - Using Statistical Tagging

Partial Parsing

- Cascading Partial Parsers
- Elementary Analysis of Grammatical Functions
  - Main Functions
  - Extracting Other Groups
- An Annotation Scheme for Groups in French
- Application: the FASTUS System
  - The Message Understanding Conferences
  - The Syntactic Layers of the FASTUS System
  - Evaluation of Information Extraction Systems

Is Syntax Necessary?

- Note
  - A parse tree is necessary to obtain the semantic representation of a sentence.
  - It is difficult to build a syntactic parser with large grammatical coverage, expensive in terms of resources, and sometimes not worth the cost.
    - Some applications need only to detect key words.
    - Other applications rely on the detection of word groups.

Word Spotting and Template Matching

- ELIZA

Table 9.1. Some ELIZA templates.

User	Psychotherapist
… I like X…	*Why do you like X?*
… I am X…	*How long have you been X?*
… father…	*Tell me more about your father*

Word Spotting and Template Matching

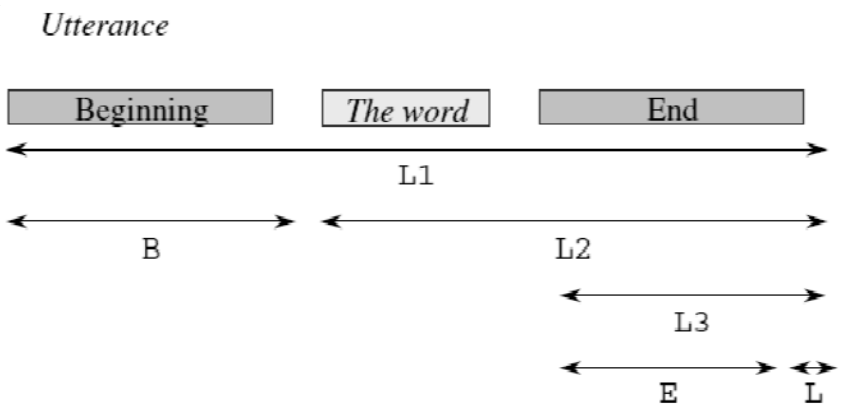
- Word Spotting in Prolog
  - utterance(U) --> beginning(B), [the_word], end(E).

  - utterance(U, L1, L) :-
    beginning(B, L1, L2),
    c(L2, the_word, L3),
    end(E, L3, L).

Word Spotting and Template Matching

- Word Spotting in Prolog
  - beginning(X, Y, Z) :- append(X, Z, Y).
    end(X, Y, Z) :- append(X, Z, Y).

Fig. 9.1. The composition of utterance.



Word Spotting and Template Matching

- Word Spotting in Prolog
  - ELIZA

ch9/ch9-eliza.pl

Multiword Detection

- Multiwords (MWE: Multiword expressions)
  - sequences of two or more words that act as a single lexical unit
  - include proper nouns (names) of persons, companies, organizations, temporal expressions describing times and dates, and numerical expressions
  - also include complex prepositions, adverbs, conjunctions, or phrasal verbs

Multiword Detection

- Multiwords (MWE: Multiword expressions)

Table 9.2. Multiwords in English and French.

Type	English	French
Prepositions **Adverbs** **Conjuntions**	*to the left hand side* *because of*	*À gauche de* *à cause de*
Names **Titles**	*British gas plc.* *Mr.Smith* *The President of the United States*	*Compagnie générale d'électricité SA* *M. Dupont* *Le président de la République*
Verbs	*give up* *go off*	*faire part* *rendre visite*

Multiword Detection

- A Standard Multiword Annotation
  - In the 1990s, the US department of defense organized a series of competitions to measure the performance of commercial and academic systems on multiword detection.
    - the Message Understanding Conferences (MUCs)
  - MUC-6 and MUC-7 defined an annotation scheme, subsequently adopted by commercial applications.

Multiword Detection

- Detecting Multiwords with Rules
  - an extension of word spotting
  - We represent multiwords with DCG rules, using variables and Prolog code to extract them from the word stream and annotate them.
  - cf. gazetteers
    - specialized dictionaries of surnames, companies, countries, and trademarks

Multiword Detection

- The Longest Match
  - Examples
    - *in front* vs. *in front of*

Table 9.3. Longer matches are preferred.

	English	**French**
Competing multiwords	*in front of* *in front*	*en face de* *en face*
Examples	*The car in front* *In front of me*	*La voiture en face* *En face de moi*

Multiword Detection

- Running the Program

ch9/ch9-multiword-ver 1.pl

```
multiword_detector(['M.', 'Dupont', was, given,
500, euros, in, front, of, the, casino], Res),
flatten(Res, Out).
```

Noun Groups and Verbs Groups

Table 9.4. Noun groups.

English	French	German
The waiter *is bringing* ***the very big dish*** *on the table*	***Le serveur*** *apporte* ***le très grand plat*** *sur* ***la table***	***Der Ober*** *bringt* ***die sehr große Speise*** *an* ***dem Tisch***
Charlotte *has eaten* ***the meal*** *of* ***the day***	***Charlotte*** *a mangé* ***le plat du Jour***	***Charlotte*** *hat* ***die Tagesspeise*** *gegessen*

Table 9.5. Verb groups.

English	French	German
The waiter ***is bringing*** *the very big dish on the table*	*Le serveur* ***apporte*** *le très grand plat sur la table*	*Der Ober* ***bringt*** *die sehr große Speise an dem Tisch*
Charlotte ***has eaten*** *the meal of the day*	*Charlotte* ***a mangé*** *le plat du Jour*	*Charlotte* ***hat*** *die Tagesspeise gegessen*

Noun Groups and Verbs Groups

- Groups Versus Recursive Phrases
  - Why word group detection?
    - A group structure is simpler and more tractable than that of a sentence.
    - Group detection uses a local strategy that can accept errors without making subsequent analyses of the rest of the sentence fail.
    - It also leaves less room for ambiguity.
    - As a result partial parsers are more precise.
      - They can capture roughly 90% of the groups successfully.

Noun Groups and Verbs Groups

- Groups Versus Recursive Phrases
  - Phrase structure rules
    - can describe group patterns
    - easier to write
    - makes the parser very fast, since there is no subgroup inside a group (no recursive phrases)
    - Finite-state automata can describe group structures.

Noun Groups and Verbs Groups

- DCG Rules to Detect Noun Groups
  - nominal([Noun | Nom]) --> noun(Noun),
    nominal(NOM).
    nominal([N]) --> noun(N).
    noun_group(NG) --> adj_group(AG), nominal(NOM),
    {append(AG, NOM, NG)}.
    noun_group(NG) --> det(D), adj_group(AG),
    nominal(NOM), {append([D|AG], NOM, NG)}.
- DCG Rules to Detect Verb Groups
- Running the Rules

Group Detection as a Tagging Problem

- Tagging Gaps
  - Example
    - [$_{NG}$ The government $_{NG}$] has [$_{NG}$ other agencies and instruments $_{NG}$] for pursuing [$_{NG}$ these other objectives $_{NG}$].

Table 9.6. Tagset to annotate noun groups.

Beginning	End	Between	No bracket (outside)	No bracket (inside)
[$_{NG}$	$_{NG}$]	$_{NG}$] [$_{NG}$	*Outside*	*Inside*

Group Detection as a Tagging Problem

- Tagging Words
  - Ramshaw and Marcus (1995) defined a tagset of three elements {I, O, B}.
  - Example
    - The/I government/I has/O other/I agencies/I and/I instruments/I for/O pursuing/O these/I other/I objectives/I ./O
- Using Symbolic Rules
- Using Statistical Tagging

Group Detection as a Tagging Problem

Table 9.7. Patterns used in the templates.

Word patterns		Noun group patterns	
Pattern	**Meaning**	**Pattern**	**Meaning**
W_0	Current word	T_0	Current noun group tag
W_{-1}	First word to left	T_{-1}, T_0	Tag bigram to left to current word
W_1	First word to right	T_0, T_1	Tag bigram to right of current word
W_{-1}, W_0	Bigram to left of current word	T_{-2}, T_{-1}	Tag bigram to left of current word
W_0, W_1	Bigram to right of current word	T_1, T_2	Tag bigram to right
W_{-1}, W_1	Surrounding words		
W_{-2}, W_{-1}	Bigram to left		
W_1, W_2	Bigram to right		
$W_{-1,-2,-3}$	Words 1 or 2 or 3 to left		
$W_{1,2,3}$	Words 1 or 2 or 3 to right		

Group Detection as a Tagging Problem

Table 9.8. The five first rules from Ramshaw and Marcus (1995).

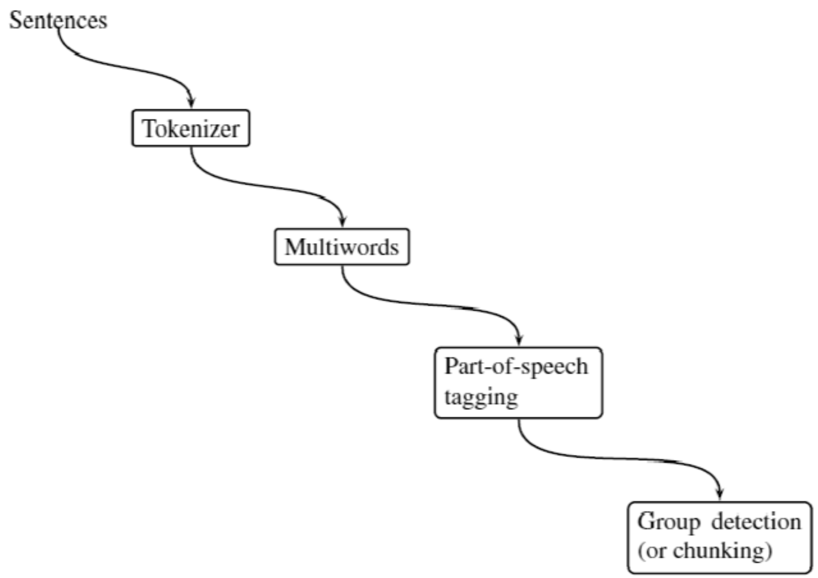
Pass	Old tag	Context	New tag
1	I	T_1 = O, P_0=JJ	O
2	-	T_{-2}=I, T_{-1}=I, P_0=WDT	B
3	-	T_{-2}=O, T_{-1}=I, P_{-1}=WDT	I
4	I	T_{-1}=I, P_0=WDT	B
5	I	T_{-1}=I, P_0=PRP	B

Group Detection as a Tagging Problem

- Using Statistical Tagging

Cascading Partial Parsers

Fig. 9.2. A cascade of partial parsers.



Remaining Issues

- Elementary Analysis of Grammatical Functions
  - Main Functions
  - Extracting Other Groups
- An Annotation Scheme for Groups in French

Application: The FASTUS System

- The Message Understanding Conferences
- The Syntactic Layers of the FASTUS System

Table 9.9. A template derived from the previous text. After Hobbs et al. (1997).

Template slots	Information extracted from the text
Incident: Date	19 Apr 89
Incident: Location	El Salvador: San Salvador (city)
Incident: Type	Bombing
Perpetrator: Individual ID	*urban guerrillas*
Perpetrator: Organization ID	*FMLN*
Perpetrator: Organization confidence	Suspected or accused by authorities: *FMLN*
Physical target: Description	*vehicle*
Physical target: Effect	Some damage: *vehicle*
Human target: Name	*Roberto Garcia Alvarado*
Human target: Description	*Attorney general: Roberto Garcia Alvarado*
Human target: Effect	*driver* *Bodyguards* Death: *Roberto Garcia Alvarado* No injury: *driver* Injury: *bodyguards*

Application: The FASTUS System

- Evaluation of Information Extraction Systems

Table 9.10. Documents in a library returned from a catalog query and split into relevant and irrelevant books.

	Relevant documents	Irrelevant documents
Retrieved	*A*	*B*
Not retrieved	*C*	*D*

Application: The FASTUS System

- Evaluation of Information Extraction Systems

Homework #3

- Proposal Idea
  - Write a 1 page proposal for the project that you will be working on for the rest of the semester.
  - It should describe what the domain is, what the problem is, how you would address the problem, and how you want the result assessed (performance measure).
- Due
  - March 24, 2009, 1pm.