



# A Statistical Grammar Model for Modern Greek

Effi Georgala

Institute for Natural Language  
Processing (IMS)

University of Stuttgart

Germany



# Motivation

---

- No previous Statistical Grammar Model for Modern Greek (MG)
- Linguistic issues of Argument Structure & Transitivity



# Outline

---

- **Why a Statistical Framework?**
- Properties of MG
- YAP Grammar for MG
- Grammar Development & Training
- Data
- Objective



# Why a Statistical Framework?

---

- **Phrase structure disambiguation:**
  - Syntactic disambiguation
  - Attachment disambiguation
- **Lexical dependencies:**
  - Subcategorization
  - Selectional preferences



# Useful Results

---

- **Linguistic theory**
- **Applications:**
  - Lexicography
  - Parsing
  - Machine Translation
  - Information Retrieval



# Outline

---

- Why a Statistical Framework?
- **Properties of MG**
- YAP Grammar for MG
- Grammar Development & Training
- Data
- Objective



# Properties of MG

---

- Pro-drop
- Rich nominal & verbal morphological system
- Agreement



# Properties of MG

---

- Relatively free word order in:
  - Clausal level
  - Phrasal level
- Cliticization & Clitic Doubling



# Properties of MG

---

- No infinitive
- 3 Moods: Indicative, Subjunctive, Imperative
- Active Participle
- 3 Negation Particles



# Outline

---

- Why a PCFG framework?
- Properties of MG
- **YAP Grammar for MG**
- Grammar Development & Training
- Data
- Objective



# YAP

---

## Grammar Rule:

### cf backbone + feature equations

SA {sframe=n\_npa;} -> NP

{agr.case=nom;agr.number=n;adjunctive=f;}

^VPA {number=n;person=3;aframe=\_npa;};



# YAP

---

## **Declaration:**

- **Categories & their Features**

```
category VPA {  
    NUMBER number;  
    PERSON person;  
    AFRAME aframe;  
};
```

- **Features & their Values**

```
enum NUMBER {sg,pl};
```



# YAP

---

## Lexicon

"βιβλιο" : NoCmNeSgNm {} ;

"βιβλιο" : NoCmNeSgAc {} ;

## Templates

VPA3 : VPA {person=3;} ;



# YAP

---

## Incorporation

VPA incorporates {number, aframe};

NP incorporates {agr.number, agr.case};

SA incorporates {saframe};

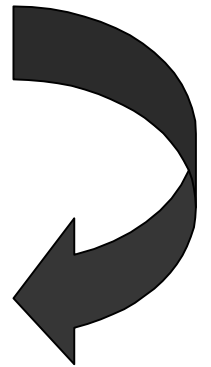


# YAP

---

## YAP rule

```
SA {sframe=n_npa;} -> NP  
{agr.case=nom;agr.number=n;adjunctive=f;}  
`VPA {number=n;aframe=_npa;};
```



## CF rules

```
SA/n_npa NP/nom/sg VPA/sg/_npa  
SA/n_npa NP/nom/pl VPA/pl/_npa
```



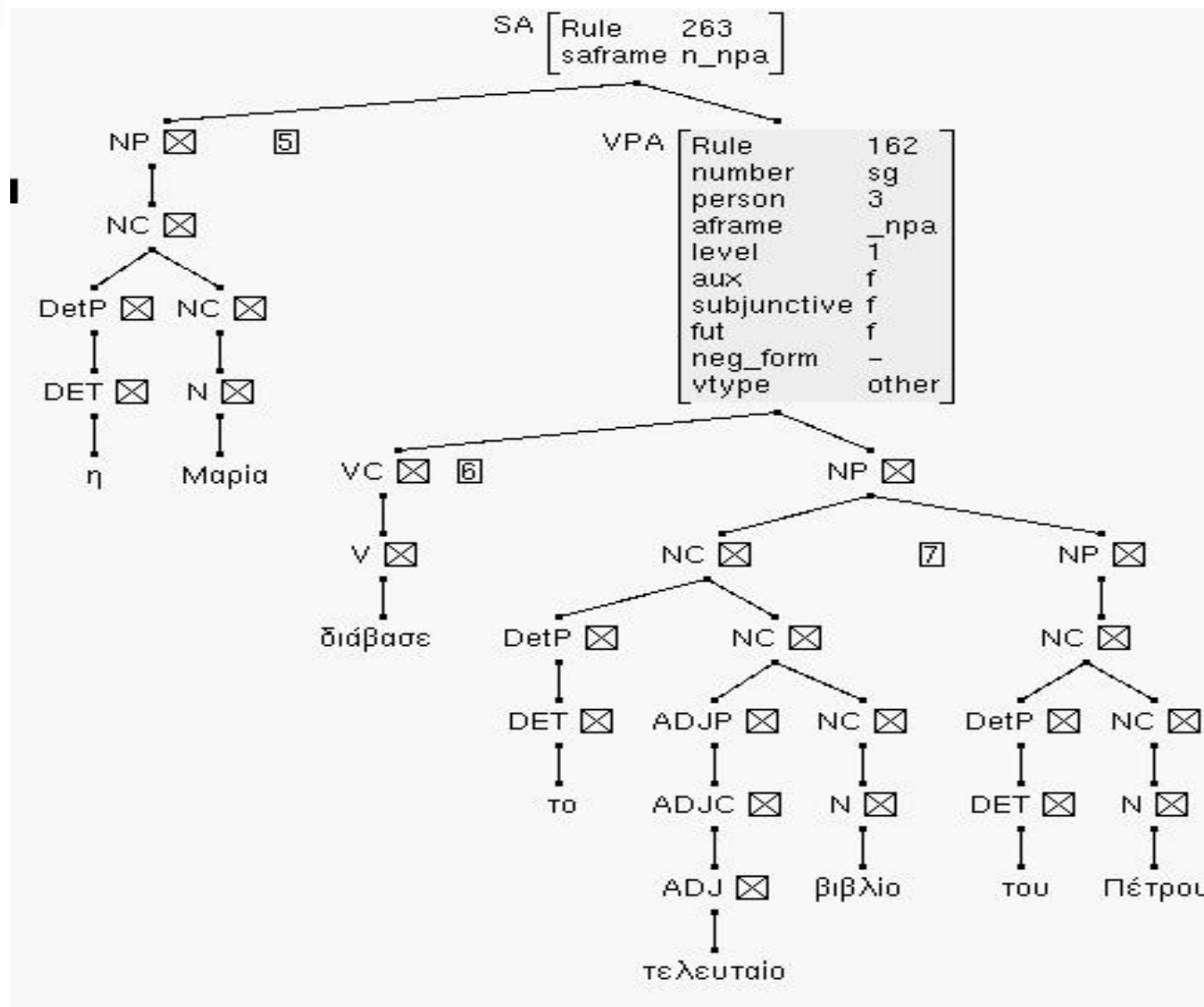
# YAP Grammar for MG

---

## Grammar rules:

- Chunk -> [Specifiers] [Modifiers] **`Head**
- Phrase -> **`Chunk** [Complements] [Adjuncts]

# YAP Grammar for MG





# YAP Grammar for MG

---

## **The Grammar covers:**

- **Fine tuned Chunks & Phrases for:**
  - Nouns
  - Adjectives
  - Adverbs
  - Prepositions
  - Verbs
- **Main & Embedded Clauses**



# YAP Grammar for MG

---

## **The Grammar models:**

- Agreement
- Appositional & predicative constructions
- Valence
- Cliticization & Clitic Doubling
- Word order in phrasal & clausal level

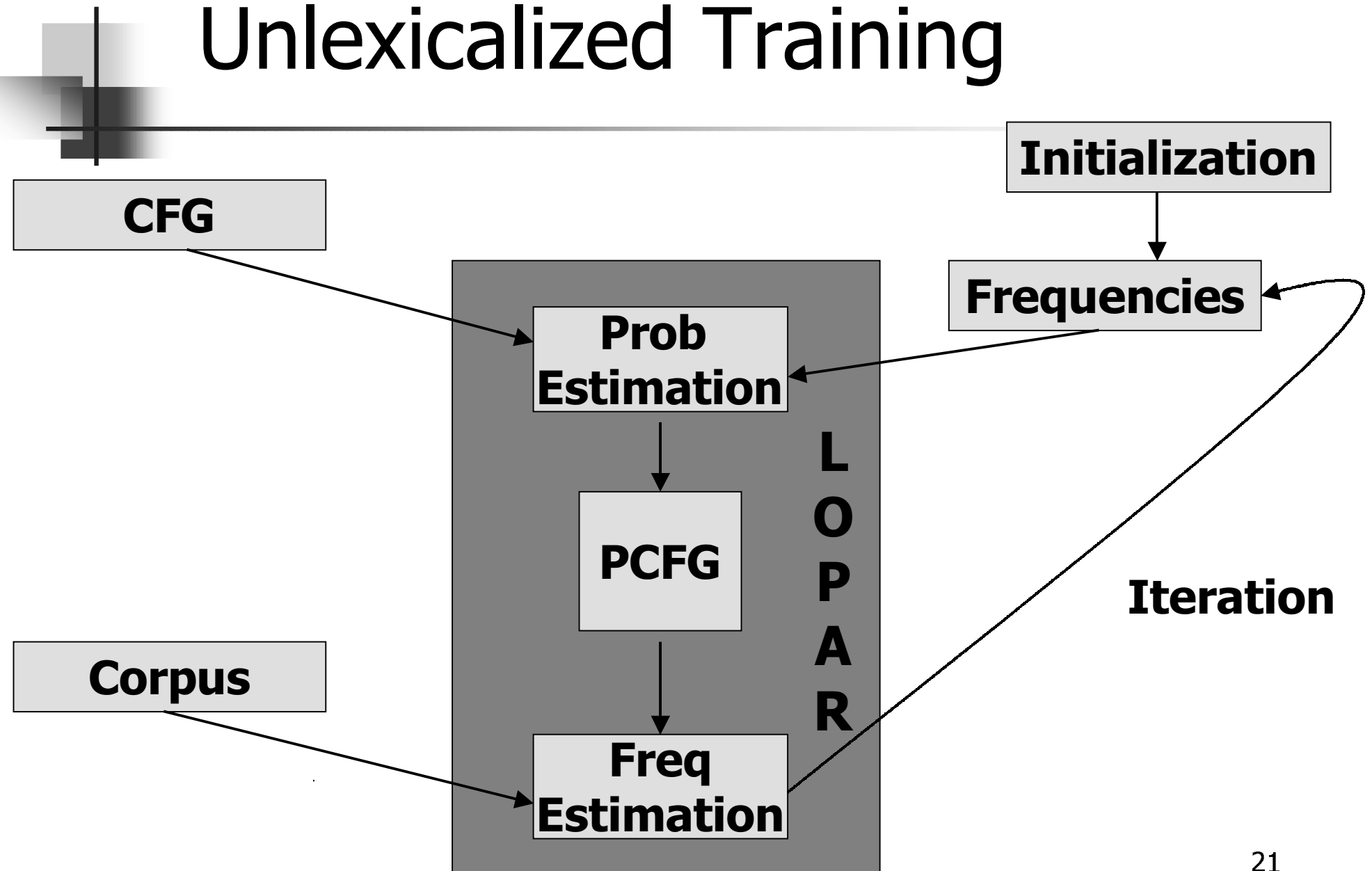


# Outline

---

- Why a PCFG framework?
- Properties of MG
- YAP Grammar for MG
- **Grammar Development & Training**
- Data
- Objective

# Unlexicalized Training





# Example

---

## **CFG-rules**

**SA/n\_npa NP/nom VPA/\_npa**

**SA/n\_npg NP/nom VPA/\_npg**



# Example

---

## **PCFG-rules before Training**

2000 SA/n\_npa NP/nom VPA/\_npa

2000 SA/n\_npg NP/nom VPA/\_npg

## **PCFG-rules after Training**

655,900 SA/n\_npa NP/nom VPA/\_npa

85,300 SA/n\_npg NP/nom VPA/\_npg



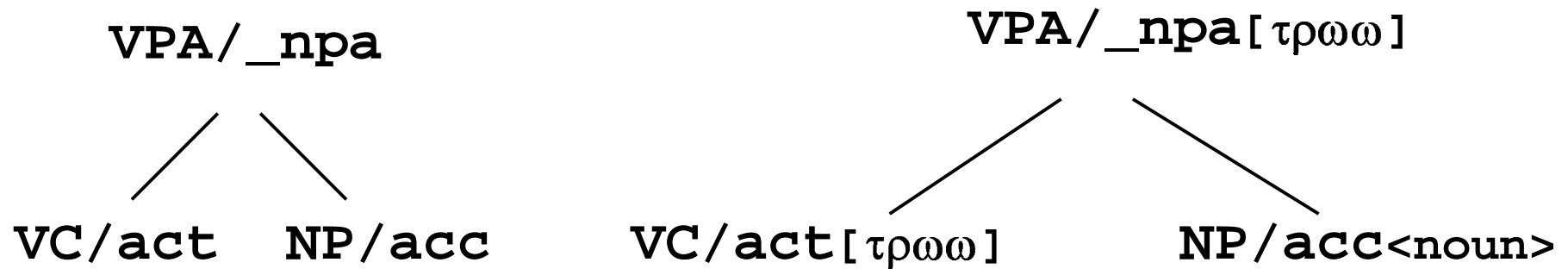
# Lexicalized Training

---

1. Lexicalization of PCFG -> HLPCFG
2. Lexicalized training of HLPCFG

# Lexicalization (example)

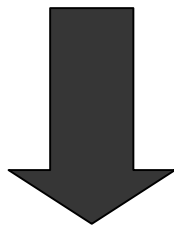
**VPA/\_npa -> VC/act' NP/acc**



# Lexicalized Training (example)

## PCFG-rule

655,900 SA/n\_npa NP/nom VPA/\_npa



## Lexicalized Rule Parameters

40,000 έχω SA/n\_npa NP/nom VPA/\_npa

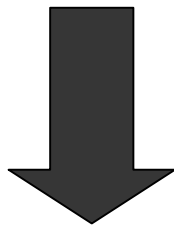
12,000 κανω SA/n\_npa NP/nom VPA/\_npa

6,500 βλέπω SA/n\_npa NP/nom VPA/\_npa

# Lexicalized Choice Parameters

## PCFG-rule

655,900 SA/n\_npa NP/nom VPA/\_npa



## Lexicalized Choice Parameters

40.00 τηλεοραση NP/acc VPA/\_npa βλεπω

12.00 αγωνα NP/acc VPA/\_npa βλεπω



# Outline

---

- Why a PCFG framework?
- Properties of MG
- YAP Grammar for MG
- Grammar Development & Training
- **Data**
- Objective



# Training Data

---

## **Newspaper Corpus** TA NEA

- 48.4 million tokens
- 2.3 million sentences

## **Processing:**

- Tokenization
- POS & Morphology Tagging
- Lemmatization



# Outline

---

- Why a PCFG framework?
- Properties of MG
- YAP Grammar for MG
- Parameter Estimation for PCFG
- Data
- **Objective**



# Objective

---

Define syntactic properties that set  
Core Transitive [Levin to appear]  
from Non Core Transitive verbs apart



# Examples

---

## **Transitive vs. Intransitive:**

Pat opened the door

Pat died

## **Core Transitive vs. Non Core Transitive:**

Pat broke my arm / \*Pat broke

Pat swept (the floor)



# Assumption

---

The contrast between Core Transitive and Non Core Transitive verbs is reflected in their argument realization



# Approach

---

- Large data samples
- Extract information on:
  - Argument realization
  - Verb alternations
  - Verb 's subcategorization options
- Cluster Analysis



## Example [Levin to appear]

---

### **Unspecified Object**

#### Sweep vs Break

Leslie swept (the floor) this morning

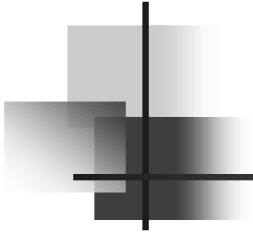
\*Kelly broke again tonight when she did  
the dishes



# Research Problems

---

- Are syntactic reflexes sufficient to distinguish between Core Transitive and Non Core Transitive verbs?
- If yes, which are the most efficient syntactic predictors for this distinction?



**END**