CMPE 590:MACHINE TRANSLATION

RESEARCH PROJECT

EUROTRA PROJECT

KADRIYE YASEMIN USTA

EUROTRA PROJECT

THE FIRST CRUSADE AGAINST THE MULTILINGUALITY PROBLEM

(1978-1993)

Agenda

01 Introduction

02 System Design

03 Computational Approach

04 Linguistic Features

05 Limitations

06 Conclusions

01

Introduction

What is EUROTRA Project?

- Expanding European Union
- Political decision to make all languages equal (NL, FR, GE, EN, DK, IT)
- GR, SP and PT joined later
- Heavy translation load (time & cost)
- EU lagging behind US and Japan in technology

01

Introduction

AIM OF EUROTRA

- To develop a prototype for MT between the languages of the European Community (Danish, Dutch, English, French, German,Greek, Italian, Portuguese and Spanish)
- To stimulate research in computational linguistics in the EC Member States.

01

Table 1: Participants of EUROTRA

Task	Group	Location
Administration	DG XII-B, CEC	Luxembourg
Danish	Copenhagen University	Copenhagen
Dutch	Rijksuniversiteit Utrecht	Utrecht
	Katholieke Universiteit Leuven	Leuven
English	UMIST	Manchester
	University of Essex	Colchester
French	University of Nancy - 2	Nancy
	University of Paris - 7	Paris
	University of Liege	Liege
German	IAI	Saarbrücken
	IKP	Bonn
Greek	Eurotra Greece	Athens
	Panepistemio tou Pethymnou	Crete
Italian	Gruppo Dima	Turin
	University of Pisa	Pisa
Portuguese	Universidade de Lisboa	Lisbon
Spanish	Universidad de Barcelona	Barcelona
	Universidad Autonoma de Madrid	Madrid
Terminology	Dublin City University	Dublin
Documentation and software clearing-house	CRETA	Luxembourg

01

Table 2: EUROTRA Workshops

Year	**Place**
1979	Aix een Provence (FR)
1980	Bangor (UK)
1981	Urbino (IT)
1982	Germersheim (DE)
1983	Cleavaux (LU)
1984	Corfu (GR)
1985	Leuven (BE)
1986	Dublin (IE)
1987	Sitges (ES)
1988	Kalø (DK)
1989	Vimeiro (PT)
1990	Noordwijkerhout (NL)
1991	St Maximin (FR)
1992	Manchester (UK)

02

System Design

< 8 >

Arhitecture Options

- Interlingua, with one analysis and one synthesis component for every language (6 languages, 12 components)
- Transfer, with rather shallow analysis and synthesis for each language and a transfer component per language pair (12 + 30 reasonably sized components)
- Direct, with a separate component for each language pair (30 components)

02

System Design

< 9 >

Arhitecture Chosen

- Transfer, with as a fundamental principle that transfer rules can only be lexical
- Considerations
  - Interlingua too hard to construct
  - Direct translation leads to much duplication (e.g. NL=>FR and NL=>EN will have many overlap)

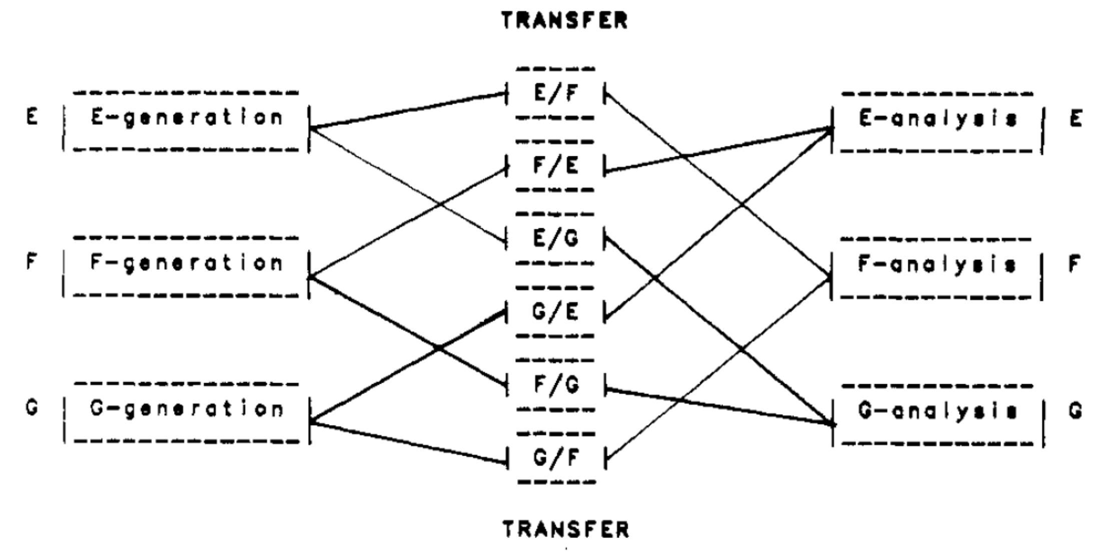
02

System Design

Transfer-based System

- The basic system:

text (L1) → IS (L1) → IS (L2) → text (L2)



02

System Design

The levels of transfer-based approach

- **ETS (Eurotra Text Structure):** The input text with formatting and publishing codes, diagrams and non-textual data
- **ENT (Eurotra Normalised Text):** The other type of input text which stripped of all non-textual data and coding
- **EMS (Eurotra Morphological Structure):** A representation of words and morphemes in word trees
- **ECS (Eurotra Constituent Structure):** A representation of syntactic constituency structure
- **ERS (Eurotra Relational Structure):** A representation of grammatical relations
- **IS (Interface Structure):** A representation of semantic dependency

03

Computational Approach

< 12 >

Computational Approach

- Unification-based grammar formalism
- Rule-writing formalism called **E-Framework**
- Two main items;
  - Objects and Structures
  - Translators and Generators

03

Computational Approach

Objects and Structures

- Each level of the representations has different objects and structures
- A defined set of features
- Structural Properties;
    - Dominance (e.g mother-daughter relationship)
    - Precedence (e.g ordering among sisters)

03

Computational Approach

Translators and Generators

- The translation is done by "translators" which take as input a single consolidated object and process it.
- The generator performs the consolidation of the unconsolidated objects that are outputted by translator.
- Three main types of rules in generator
    - Structure-building Rules
    - Feature Rules
    - Filter Rules

03

Computational Approach

< 15 >

Structure-building Rules

- Two types of it
  - **B-rules ('b' for ‘building’):** creates a new node
  - **L-rules ('l' for 'leaf') :** atomic dictionary entries

03

Computational Approach

Feature Rules

- Two types of it
  - **I-rules ('i' for 'insertion'):** used for insertion
  - **F-rules ('f' for 'feature') :** defining a feature to the structure

03

Computational Approach

Filter Rules

- Two types of it
  - **S-rules ('s ' for 'strict'):** specify acceptability conditions for structures
  - **K-rules ('k' for 'killer'):** suppress illegal or unwanted structures.

Example Linguistic Analysis

- Input, in Eurotra ETS or ENS:

Die Industrie kennt dieses Problem seit einiger Zeit.
"Industry has known about this problem for some time"(English)

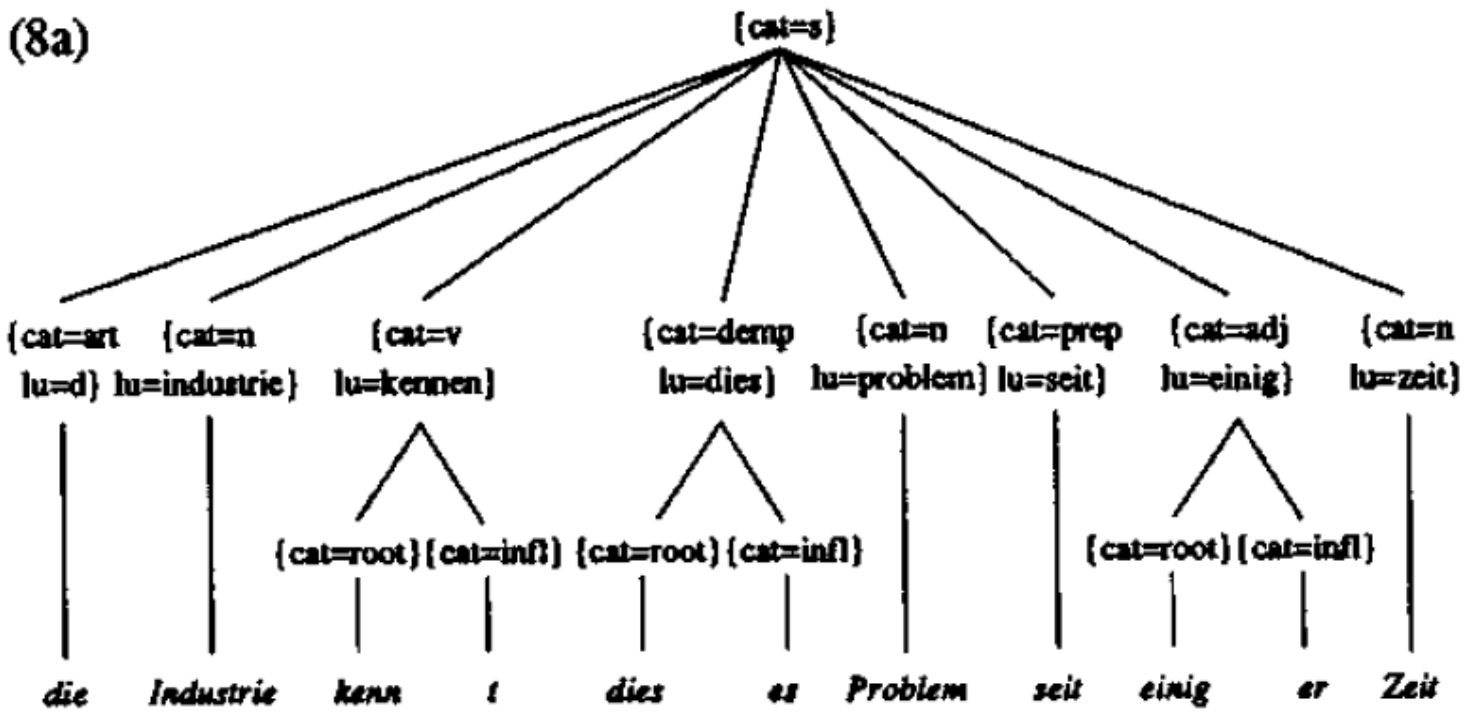
04

Linguistic Features

Example Linguistic Analysis

Creating Morphology Tree - EMS

(8a)



04

Linguistic Features

Example Linguistic Analysis

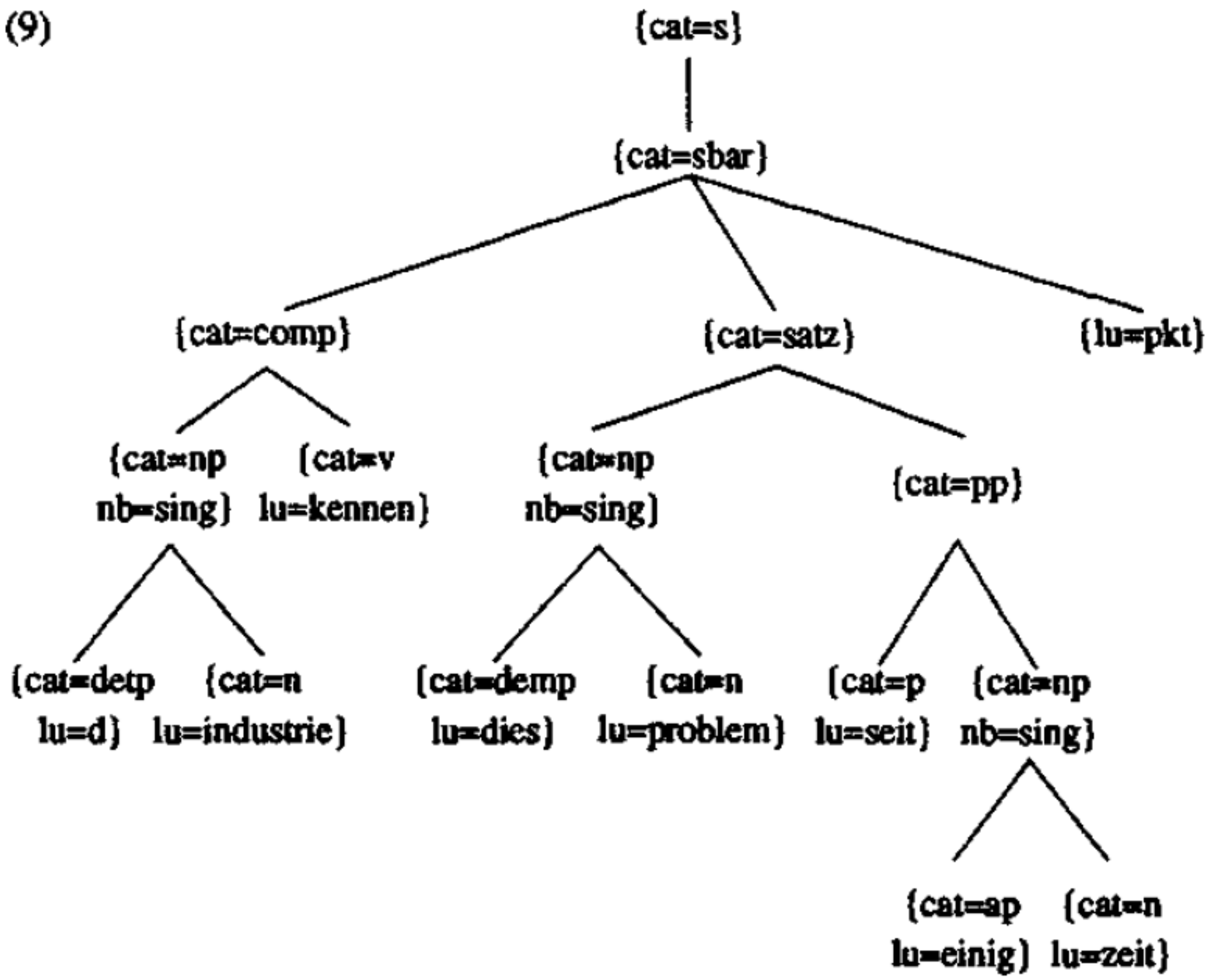
- Creating Morphology Rules - EMS

```
{cat=s}
 <{cat=art, lu=d, msdefs=msdef, gender=fem,
   nb=sing},
  {cat=n, lu=industrie, gender=fem, nb=sing},
  {cat=v, lu=kennen, mstense=pres, nb=sing,
   pers=3}
   <{cat=root, lu=kenn},
    {cat=infl, lu=t}>,
  {cat=art, lu=dies, msdefs=msdef, gender=neut,
   nb=sing},
   <{cat=root, lu=dies},
    {cat=infl, lu=es}>,
  {cat=n, lu=problem, gender=neut, nb=sing},
  {cat=prep, lu=seit, ....},
  {cat=adj, lu=einig, ....},
   <{cat=root, lu=einig},
    {cat=infl, lu=er}>,
  {cat=n, lu=zeit, gender=fem, nb=sing}>
```

Example Linguistic Analysis

- Phrase Analysis, creating tree of phrases - ECS

(9)



04

Linguistic Features

Example Linguistic Analysis

- Phrase Analysis, creating rules for phrases - ECS

```
np = {cat=np, case=C, nb=N}
      [^{cat=detp, case=C, nb=N, gender=G},
       *{cat=ap, case=C, nb=N, gender=G},
       {cat=n, case=C, nb=N, gender=G}].
```

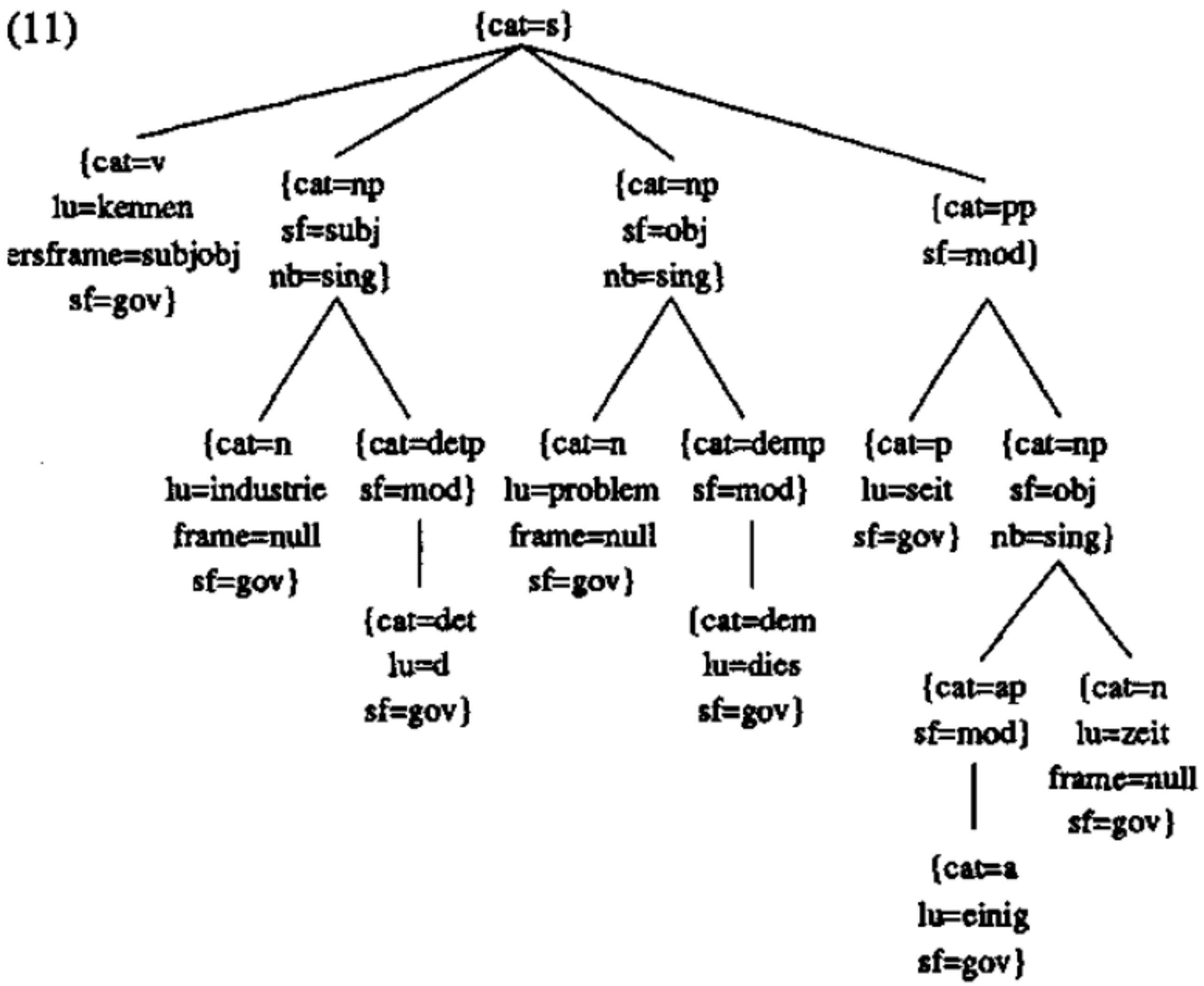
04

Linguistic Features

Example Linguistic Analysis

- Dependency Grammar Analysis, creating tree - ERS

(11)



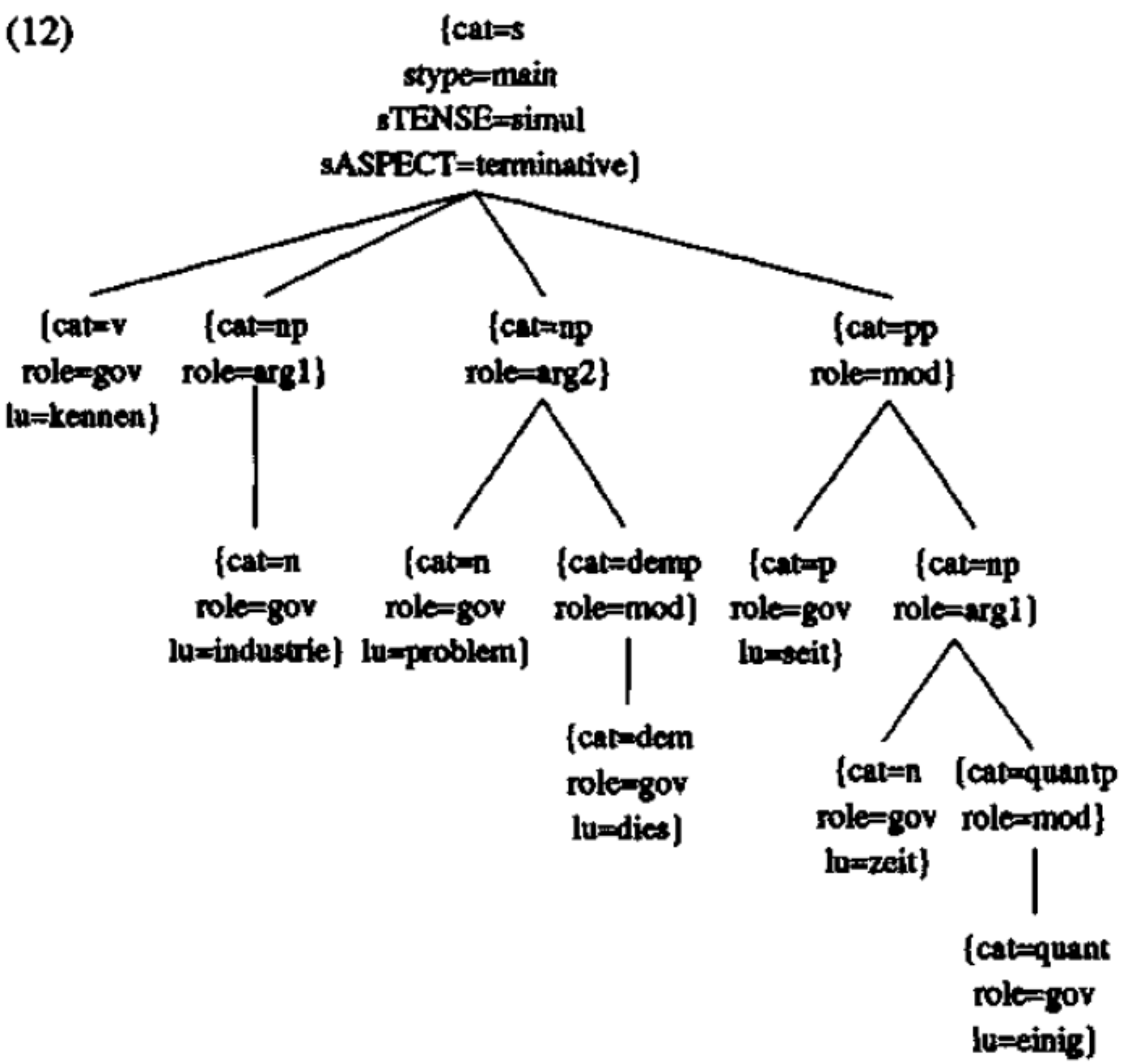
04

Linguistic Features

Example Linguistic Analysis

- Creating Interface Structure - IS

(12)



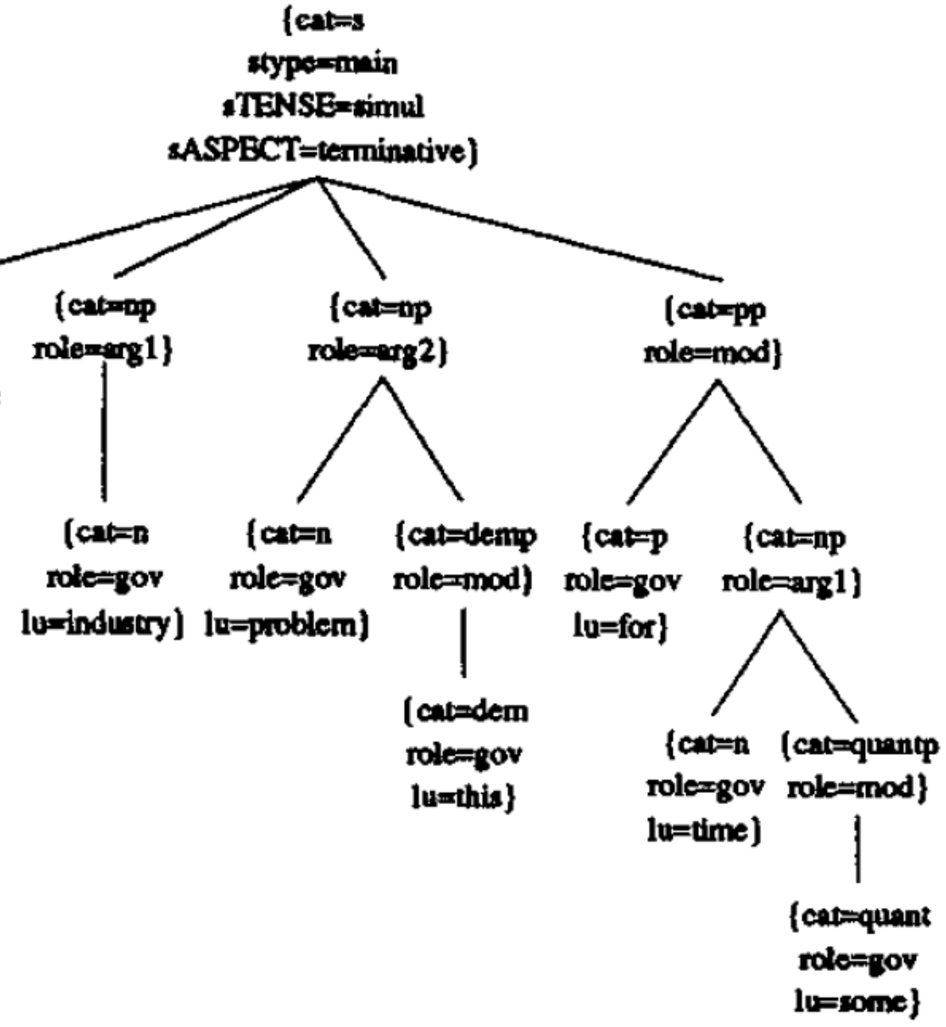
04

Linguistic Features

Example Linguistic Analysis

- Translating to English Interface Structure - IS

(13a)



04

Linguistic Features

Biggest MT problems

- Ambiguity (not solvable with linguistic knowledge alone)
- Computational complexity
- Robustness (react sensibly to unexpected or ill-formed input)
- Evaluation

05

Limitations

Biggest Implementation problems

- Hard to start building something before it has invented
- Different scientific backgrounds and participation motives
- Hard to invent something with a crowd of 300
- Too many interdependencies between the teams

05

Limitations

- The most ambitious MT project :)

- The project did not deliver what was promised

- The impact was enormous
    - The project put Europe on the international NLP map.
    - some starting points for later MT activities in some languages (Danish, German)
    - De facto network of institutes and individuals all over Europe
    - Clear lessons for later EU R&D programmes

06

Conclusions

THANKS FOR LISTENING