CMPE 590: MACHINE TRANSLATION

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
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
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.
<table>
<thead>
<tr>
<th>Task</th>
<th>Group</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administration</td>
<td>DG XII-B, CEC</td>
<td>Luxembourg</td>
</tr>
<tr>
<td>Danish</td>
<td>Copenhagen University</td>
<td>Copenhagen</td>
</tr>
<tr>
<td>Dutch</td>
<td>Rijksuniversiteit Utrecht</td>
<td>Utrecht</td>
</tr>
<tr>
<td>Dutch</td>
<td>Katholieke Universiteit Leuven</td>
<td>Leuven</td>
</tr>
<tr>
<td>English</td>
<td>UMIST</td>
<td>Manchester</td>
</tr>
<tr>
<td>English</td>
<td>University of Essex</td>
<td>Colchester</td>
</tr>
<tr>
<td>French</td>
<td>University of Nancy - 2</td>
<td>Nancy</td>
</tr>
<tr>
<td>French</td>
<td>University of Paris - 7</td>
<td>Paris</td>
</tr>
<tr>
<td>French</td>
<td>University of Liege</td>
<td>Liege</td>
</tr>
<tr>
<td>German</td>
<td>IAI</td>
<td>Saarbrücken</td>
</tr>
<tr>
<td>German</td>
<td>IKP</td>
<td>Bonn</td>
</tr>
<tr>
<td>Greek</td>
<td>Eurotra Greece</td>
<td>Athens</td>
</tr>
<tr>
<td>Greek</td>
<td>Panepistemio tou Pethymnou</td>
<td>Crete</td>
</tr>
<tr>
<td>Italian</td>
<td>Gruppo Dima</td>
<td>Turin</td>
</tr>
<tr>
<td>Italian</td>
<td>University of Pisa</td>
<td>Pisa</td>
</tr>
<tr>
<td>Portuguese</td>
<td>Universidade de Lisboa</td>
<td>Lisbon</td>
</tr>
<tr>
<td>Spanish</td>
<td>Universidad de Barcelona</td>
<td>Barcelona</td>
</tr>
<tr>
<td>Spanish</td>
<td>Universidad Autonoma de Madrid</td>
<td>Madrid</td>
</tr>
<tr>
<td>Terminology</td>
<td>Dublin City University</td>
<td>Dublin</td>
</tr>
<tr>
<td>Documentation and software clearing-house</td>
<td>CRETA</td>
<td>Luxembourg</td>
</tr>
<tr>
<td>Year</td>
<td>Place</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------------------------</td>
<td></td>
</tr>
<tr>
<td>1979</td>
<td>Aix een Provence (FR)</td>
<td></td>
</tr>
<tr>
<td>1980</td>
<td>Bangor (UK)</td>
<td></td>
</tr>
<tr>
<td>1981</td>
<td>Urbino (IT)</td>
<td></td>
</tr>
<tr>
<td>1982</td>
<td>Germersheim (DE)</td>
<td></td>
</tr>
<tr>
<td>1983</td>
<td>Cleavaux (LU)</td>
<td></td>
</tr>
<tr>
<td>1984</td>
<td>Corfu (GR)</td>
<td></td>
</tr>
<tr>
<td>1985</td>
<td>Leuven (BE)</td>
<td></td>
</tr>
<tr>
<td>1986</td>
<td>Dublin (IE)</td>
<td></td>
</tr>
<tr>
<td>1987</td>
<td>Sitges (ES)</td>
<td></td>
</tr>
<tr>
<td>1988</td>
<td>Kalø (DK)</td>
<td></td>
</tr>
<tr>
<td>1989</td>
<td>Vimeiro (PT)</td>
<td></td>
</tr>
<tr>
<td>1990</td>
<td>Noordwijkerhout (NL)</td>
<td></td>
</tr>
<tr>
<td>1991</td>
<td>St Maximin (FR)</td>
<td></td>
</tr>
<tr>
<td>1992</td>
<td>Manchester (UK)</td>
<td></td>
</tr>
</tbody>
</table>
Architecture 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)
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)
Transfer-based System

- The basic system:

\[
\text{text (L1) \rightarrow IS (L1) \rightarrow IS (L2) \rightarrow text (L2)}
\]
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
Computational Approach

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

03

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)
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
Structure-building Rules

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

Creating Morphology Tree - EMS

(8a)

```
die Industrie kennt dies es Problem seit einig er Zeit
```
Example Linguistic Analysis

- Creating Morphology Rules - EMS

```plaintext
{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
Example Linguistic Analysis

- Phrase Analysis, creating rules for phrases - ECS

\[ np = \{ \text{cat}=np, \text{case}=C, \text{nb}=N \} \]
\[ ^{\{ \text{cat}=\text{detp}, \text{case}=C, \text{nb}=N, \text{gender}=G \},} \]
\[ ^{\{ \text{cat}=\text{ap}, \text{case}=C, \text{nb}=N, \text{gender}=G \},} \]
\[ ^{\{ \text{cat}=\text{n}, \text{case}=C, \text{nb}=N, \text{gender}=G \}].} \]
Example Linguistic Analysis

- Dependency Grammar Analysis, creating tree - ERS

(11)

```
{cat=v
lu=kennen
crsframe=subobj
sf=gov}

{cat=np
sf=subj
nb=sing}  {cat=np
sf=obj
nb=sing}  {cat=pp
sf=mod}

{cat=dtp
lu=industrie
frame=null
sf=gov}

{cat=n
lu=problem
frame=null
sf=gov}  {cat=demp
lu=zeit
frame=null
sf=gov}

{cat=dct
lu=d
sf=gov}  {cat=dem
lu=dies
sf=gov}

{cat=sp
sf=mod}  {cat=n
lu=sein
sf=gov}

{cat=a
lu=cinig
sf=gov}
```
Example Linguistic Analysis

- Creating Interface Structure - IS

(12)

```
{cat=s
  stype=main
  sTENSE=simul
  sASPECT=terminative)

{cat=v
  role=gov
  lu=kennen}

{cat=np
  role=arg1}

{cat=np
  role=arg2}

{cat=pp
  role=mod}
```

```
{cat=n
  role=gov
  lu=industrie}

{cat=n
  role=gov
  lu=problem}

{cat=demp
  role=mod
  lu=ziel}

{cat=p
  role=gov
  lu=zeit}

{cat=nm
  role=arg1
  lu=einig}

{cat=quant
  role=gov
  lu=einig}
```
Example Linguistic Analysis

- Translating to English Interface Structure - IS
Biggest MT problems

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