Logic Programming Infrastructure for Inferences on FrameNet

Peter Baumgartner
MPI Informatik, Saarbrücken

Aljoscha Burchardt
Universitaet des Saarlandes, Saarbruecken
Motivation
Motivation

Problem 1: Low Precision

Problem 2: Low Recall
(Miss highly relevant page returned for search term BMW buy Rover)
Approaches

- Stemming (buys – buy)
- Synonyms (purchase – buy), e.g. from WordNet
- Compare words on page and words of query term

Statistics based methods mostly help to improve recall

Idea of our Approach: Combination (roughly):

- Linguistic methods
  state-of-the-art parsing, synonyms, feature structures, word sense disambiguation

  -> High recall

- Logic based method
  Structured, frame-like representation
  Reasoning with background knowledge

  -> high precision

Next: knowledge representation framework

Logic Programming Infrastructure for Inferences on FrameNet
From Natural Language Text to Frame Representation

FrameNet
550 Frames
7000 Lex Units

Text

Linguistic Method

Buy Rover

Sell BA Rover

Frame Representation
Com GT
Buyer: BMW
Seller: BA
Goods: Rover
Money: unknown

Logic

Deduction System

Logic Programming Infrastructure for Inferences on FrameNet
Representing Text as Logical Facts

_BMW bought Rover from BA_

_Assumption:_ linguistic method delivers:

```
buy1: buy
    buyer: "BMW"
    goods: "Rover"
```

... as a Logic Program (facts):

```
buy(buy1).
buyer(buy1,"BMW").
goods(buy1,"Rover").
```

**How to realize this task:**
- Linguistic method knows about "basic" FrameNet Frames (those admitting linguistic realization)
- Lexical units of FrameNet frames backed up by WordNet Synonym sets
- Mapping of parse trees to frames can be learned

**Extension:** some parsers (Xerox LFG parser) deliver additional valuable information, e.g. that BMW is a manufacturer
**A First Application: Transfer of Role Fillers**

(Slide by Gerd Fliedner)

<table>
<thead>
<tr>
<th>FrameNet</th>
<th>Request</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Auftrag (noun)</strong></td>
<td><strong>Message</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Receiving</th>
<th>erhalten (verb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recipient</td>
<td>Donor</td>
</tr>
<tr>
<td></td>
<td>Theme</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PRoDS</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DSub</td>
<td>Flugzeughersteller (noun)</td>
<td>PPMod von (prem)</td>
</tr>
<tr>
<td>DObj</td>
<td></td>
<td>DObj Auftrag (noun)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Topologie &amp; Chunker</th>
<th>WF</th>
<th>LK</th>
<th>KS</th>
<th>MF</th>
<th>RK (simple)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEF-ART</td>
<td>S</td>
<td>PRAEF</td>
<td>EIGEN</td>
<td>DEF-ART</td>
<td>S</td>
</tr>
<tr>
<td>Text</td>
<td>Der Flugzeughersteller hat von Großbritannien den Auftrag für 25 Transportflugzeuge erhalten.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Morph</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>der (PRON, ART)</td>
<td>Flugzeughersteller (S)</td>
<td>haben (V)</td>
<td>von (PRAEF, EIGEN)</td>
<td>Großbritannien (S)</td>
<td>der (PRON, ART)</td>
</tr>
<tr>
<td>die (PRON, ART)</td>
<td>Flugzeughersteller (S)</td>
<td></td>
<td></td>
<td>der (PRON, ART)</td>
<td>Auftrag (S)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>für (ADV, PRAEF)</td>
<td>25 (NUM)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Transportflugzeug (S)</td>
<td>erhalten (AUX, ART)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(PUNCT)</td>
</tr>
</tbody>
</table>

The plane manufacturer has from Great Britain the order for 25 transport planes received.

"Challenge": Fill in the missing elements of „Request“ frame
Transfer of Role Fillers

The plane manufacturer has from Great Britain the order for 25 transport planes received.

Parsing gives partially filled FrameNet frame instances of „receive“ and „request“:

receive1: receive
  target: „received“
  donor: „Great Britain“
  recipient: manufacturer1
  theme: request1

request1: request
  target: „order“
  speaker: „Great Britain“
  addressee: manufacturer
  message: „transport plane“

- Transfer of role fillers done so far manually
- Can be done automatically. By „model generation“
Computing Models with KRHyper

- Disjunctive logic programs
- Stratified default negation
- Perfect model semantics
- Also stable models, possible models
- Serious implementation (OCaml)

\[ a \]
\[ b; c : a \]
\[ a; d : c \]
\[ \text{false} : a, b \]
\[ e : c, \text{not } d \]

\[
\begin{array}{c}
\{\} \not\models (1) \\
\{a\} \not\models (2) \\
\{a, b\} \not\models (4) X \\
\{a, c\} \models (1)-(4)
\end{array}
\]

- Variant for predicate logic
- Extensions: minimal models, abduction, **default negation**
Transfer of Role Fillers

The plane manufacturer has from Great Britain the order for 25 transport planes received.

Parsing gives partially filled FrameNet frame instances of „receive“ and „request“:

receive1: receive
  target: „received“
  donor: „Great Britain“
  recipient: manufacturer1
  theme: request1

request1: request
  target: „order“
  speaker: „Great Britain“
  addressee: manufacturer
  message: „transport plane“

- Transfer of role fillers done so far manually
- Can be done automatically. By „model generation“
Transfer of Role Fillers by Logic Programming

receive1:

```
receive

target: „received“

 donor: „Great Britain“

 recipient: manufacturer1

 theme: request1
```

request1:

```
request

target: „order“

 speaker: „Great Britain“

 addressee: 

 message: „transport plane“
```

Rules

- speaker(Request, Donor) :-
  receive(Receive),
  donor(Receive, Donor),
  theme(Receive, Request),
  request(Request).

Facts

- receive(receive1).
- donor(receive1, „Great Britain“).
- theme(receive1, request1).
- request(request1).
Translation of FrameNet Frames to Logic Programs

So far:

- logical representation of analyzed text, and
- impression of usefulness of logic programming approach
  using hand-crafted "expensive" rules. But:

FrameNet offers useful information that can be translated
once and for all into a logic program, in a systematic way:

- Inheritance among frames
- "Uses" relationships (partial inheritance)
- "Subframe" relationships

This way, FrameNet is equipped with a formal semantics!

Also realized in our translation: default values

Next: default values in some detail, "Uses" relationship in brief
Default Values

Insert default value as a role filler in absence of specific information

Example:

receive1:

<table>
<thead>
<tr>
<th>receive</th>
</tr>
</thead>
<tbody>
<tr>
<td>target: „received“</td>
</tr>
<tr>
<td>donor: „Great Britain“</td>
</tr>
<tr>
<td>recipient: manufacturer1</td>
</tr>
<tr>
<td>theme: request1</td>
</tr>
</tbody>
</table>

request1:

<table>
<thead>
<tr>
<th>request</th>
</tr>
</thead>
<tbody>
<tr>
<td>target: „order“</td>
</tr>
<tr>
<td>speaker: „Great Britain“</td>
</tr>
<tr>
<td>addressee:</td>
</tr>
<tr>
<td>message: „transport plane“</td>
</tr>
</tbody>
</table>

Should transfer "donor" role filler only if "speaker" is not already filled:

default_request_speaker(Request, Donor) :-
  receive(Receive),
  donor(Receive, Donor),
  theme(Receive, Request),
  request(Request).
Default Values

Insert default value as a role filler in absence of specific information

Example:
In Stock Market context use default "share" for "goods" role of "buy":

    default_buy_goods(Buy, "share") :-
    'Buy is an event in a stock market context'.

Example:
Disjunctive (uncertain) information

Linguistic analysis is uncertain whether "Rover" or "Chrysler" was bought:

    default_buy_goods(buy1,"Rover").
    default_buy_goods(buy1,"Chrysler").

This amounts to two models, representing the uncertainty
They can be analyzed further
Default Values

Insert default value as a role filler in absence of specific information

Example:
Generic "typed" default value:

\[
\text{default_commerce_goods_transfer_money}(\_, \text{unspecified\_money}).
\]

Generic default value, general scheme:

\[
\text{default}_F_R(\_, \text{unspecified}).
\]

where \( F \) is a Frame with role \( R \)

Note:

Apply general scheme only to basic frames, but omit FEE role.
Otherwise every frame will be filled right away, which is pointless!
Default Value – General Transformation

Technique:

\[
\begin{align*}
a & : - \text{ not not}_a. \\
\text{not}_a & : - \text{ not a.}
\end{align*}
\]

has two stable models: one where a is true and one where a is false

Choice to fill with default value or not:

\[
\text{goods}(F,R) : - \\
\text{not not}_\text{goods}(F,R), \\
\text{buy}(F), \\
\text{default}_\text{buy}_\text{goods}(F,R).
\]

\[
\text{not}_\text{goods}(F,R) : - \\
\text{not goods}(F,R), \\
\text{buy}(F), \\
\text{default}_\text{buy}_\text{goods}(F,R).
\]

Case of waiving default value:

\[
\text{false} : - \\
\text{buy}(F), \\
\text{default}_\text{buy}_\text{goods}(F,R), \\
\text{goods}(F,R), \\
\text{goods}(F,R), \\
\text{not equal}(R1,R2).
\]

\[
\text{equal}(X,X).
\]

Require at least one filler for role:

\[
\text{false} : - \\
\text{buy}(F), \\
\text{not some}_\text{buy}_\text{goods}(F).
\]

Role is filled:

\[
\text{some}_\text{buy}_\text{goods}(F) : - \\
\text{buy}(F), \\
\text{goods}(F,R).
\]
The "Uses" Relation

Pragmatics: offer different perspectives on Frames

```plaintext
buy1: commerce_goods_transfer
    buyer: "BMW"
    seller: "BA"
    goods: "Rover"
    money: unspecified_money
```

Technically: partial inheritance
The "Uses" Relation – Partial Inheritance

"Upwards" inheritance:

buy1: commerce_goods_transfer
    buyer: "BMW"
    seller: "BA"
    goods: "Rover"
    money: unspecified_money

- Create instance of "used" frame
- Transfer role fillers of "using" frame
- Use default values for extra roles of "used" frame
The "Uses" Relation – Partial Inheritance

"Upwards" inheritance – slightly different scenario:

- Create instance of "used" frame
- transfer role fillers of "using" frame
- Use default values for extra roles of "used" frame
The "Uses" Relation – Partial Inheritance

"Downwards " inheritance:

cgt1: commerce_goods_transfer
    buyer: "BMW"
    seller: "BA"
    goods: unspecified
    money: unspecified_money

- Create instances of "using" frames
- Transfer role fillers of "used" frame

Note: "money" role is not inherited

Rules accomplishing partial inheritance can be derived automatically!
Query Evaluation

Text

BMW bought Rover from BA

Linguistic Method

FrameNet

Com GT

Buy

Sell

BMW Rover

BA Rover

L. Pr.

KRHyper

Answer

buy(buy1).
buyer(buy1,"BMW").

KRHyper

Query:
"Who bought Rover?"

Logic Programming Infrastructure for Inferences on FrameNet
Query Evaluation - Technical

Who bought Rover from whom?

As a conjunctive query:

\[
Q: \quad \text{solution}(	ext{Buyer}, \text{Seller}) :\quad \text{commerce\_goods\_transfer}(E), \text{buyer}(E, \text{Buyer}), \text{goods}(E, "Rover"), \text{seller}(E, \text{Seller}).
\]

Assume models of text \(M_1, \ldots, M_n\) already computed

Different Reasoning Tasks

Credulous: exists \(M_i\) such that \(M_i \cup \{Q\} \models \text{solution}(B, S)\) ?

(for some \(B\) and \(S\))

Skeptical: for all \(M_i\), does \(M_i \cup \{Q\} \models \text{solution}(B, S)\) hold?

(for some \(B\) and \(S\))

Both can be solved by inspecting models of \(M_i \cup \{Q\}\)
Next Step: RTE Challenge

- Recognizing Text Entailment Challenge
  Bar Ilan University, Israel, November 2004
- Compare natural language processing systems for IR, QA, ... on a common test set
- "Textual entailment problem":
  - Given: a text snippet "text"
  - Given: a text snippet "hypothesis"
  - Question: Does "text" entail "hypothesis"?
- 300 sample pairs available now as a test set, total 1000
- Test goes beyond word sense disambiguation and beyond named entity recognition. Need "semantic" processing
- Challenge is considered as difficult by its creators
- Our approach:
  - compute models of "text" and "hypothesis"
  - compare models using further background knowledge
## RTE Challenge - Examples

<table>
<thead>
<tr>
<th>Text</th>
<th>Hypothesis</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Doug Lawrence <em>bought</em> the impressionist oil landscape by J. Ottis Adams in the mid-1970s at a Fort Wayne antiques dealer</td>
<td>Doug Lawrence <em>sold</em> the impressionist oil landscape by J. Ottis Adams</td>
<td>False</td>
</tr>
<tr>
<td>Eyeing the huge market potential, currently led by Google, Yahoo <em>took over</em> search company Overture Services Inc last year</td>
<td>Yahoo <em>bought</em> Overture.</td>
<td>True</td>
</tr>
<tr>
<td>The market value of u.s. overseas assets <em>exceeds</em> their book value.</td>
<td>The market value of u.s. overseas assets <em>equals</em> their book value.</td>
<td>False</td>
</tr>
<tr>
<td>Crude oil for April delivery <em>traded</em> at $37.80 a barrel, <em>down</em> 28 cents</td>
<td>Crude oil <em>prices rose</em> to $37.80 per barrel</td>
<td>False</td>
</tr>
<tr>
<td>Guerrillas killed a <em>peasant</em> in the city of Flores</td>
<td>Guerrillas killed a <em>civilian</em></td>
<td>True</td>
</tr>
<tr>
<td>Clinton's new book is <em>not</em> big seller here</td>
<td>Clinton's book is a <em>big seller</em></td>
<td>False</td>
</tr>
</tbody>
</table>
Conclusions

Summary

- Propose model computation paradigm for "semantical" processing of natural language text
- Target application: information retrieval from templates, question answering
- Builds on readily developed FrameNet ontology

Lots of Open Ends

- Implementation (Master's thesis, in progress)
- Practical evaluation, in particular RTE challenge
- Negation, Anaphora resolution
- Background knowledge: combination with ontologies like SUMO
- Relevance of proposed framework for Semantic Web