Challenges in Ontology learning with focus on crowd sourcing

Student: Luka Bradeško
Advisor: Dunja Mladenčić
Outline

- Motivation + Short Curious Cat DEMO
- Introduction to knowledge acquisition
- Knowledge Acquisition Challenges
- Crowdsourcing Challenges
- Conversational Clients Challenges
Motivation

• We need knowledge to be able to solve complex every day tasks
• Knowledge is important for understanding and adjusting to every day situations.
• No computer has yet the commonsense of a six year child.
Motivation

• Use cases for knowledge bases:
  – Complex information retrieval:

Reasoning-based, question answering
Motivation

- Use cases for knowledge bases:
  - Decision support using reasoning
  - How to transport different types of products (Euridice EU project)
Motivation

In Collaboration with Cycorp
Knowledge acquisition

• Constructing a knowledge base is expensive
  – The Cyc KB was mostly manually constructed over the last 25 years

• Coupling information extraction and knowledge acquisition lets us construct a knowledge base with no or little human interaction
Information extraction and knowledge acquisition

- Knowledge acquisition: **Adding extracted information into a knowledge base**

![Diagram showing the process of knowledge acquisition withCrowdsourcing](image)
Textual Data approach

IE System

- Seeds
- Corpus
- Theory

Text pre-processing
- Machine learning
- Using text or seed ontology
- Extract patterns
- Information Extraction
- Learn new patterns
- Assesment

Adds extracted information to the theory
Crowdsourcing approach

Legend:
- Initialization
- KA loop
Knowledge acquisition

• Different approaches
  – By Prior Knowledge
    • Using pre-existing ontology
    • Starting from scratch
  – By input
    • Structured data
    • Semi structured data
    • Unstructured data
    • Crowdsourcing
Knowledge acquisition

• Different approaches
  – By Prior Knowledge
    • Using pre-existing ontology
    • Starting from scratch
  – By input
    • Structured data
    • Semi structured data
    • Unstructured data
    • Social Crowdsourcing
      Using conversational clients
Semantic Chat: Social Conversational Crowdsourcing

<table>
<thead>
<tr>
<th>Questions</th>
<th>Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>Machine</td>
</tr>
<tr>
<td>Machine</td>
<td>Human</td>
</tr>
</tbody>
</table>

- NL Generation (Cyc)
- Patterns
- Predefined options (menus)
- Knowledge guided input
- Free text input
## Semantic Chat: Social Conversational Crowdsourcing

<table>
<thead>
<tr>
<th>Questions</th>
<th>Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>Machine</td>
</tr>
<tr>
<td>Machine</td>
<td>Human</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KB</th>
<th></th>
<th></th>
</tr>
</thead>
</table>

- NL Generation (Cyc)
- Patterns
- Predefined options (menus)
- Knowledge guided input
- Free text input
# Semantic Chat: Social Conversational Crowdsourcing

<table>
<thead>
<tr>
<th>Questions</th>
<th>Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>Human</td>
</tr>
<tr>
<td>Machine</td>
<td>Machine</td>
</tr>
</tbody>
</table>

- NL Generation (Cyc)
- Patterns

- Predefined options (menus)
- Knowledge guided input
- Free text input
Semantic Chat: Social Conversational Crowdsourcing

<table>
<thead>
<tr>
<th>Questions</th>
<th>Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>Machine</td>
</tr>
<tr>
<td>Machine</td>
<td>Human</td>
</tr>
</tbody>
</table>

KB

- NL Generation (Cyc)
- Patterns

- Predefined options (menus)
- Knowledge guided input
- Free text input
Semantic Chat: Social Conversational Crowdsourcing

<table>
<thead>
<tr>
<th>Questions</th>
<th>Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>Human</td>
</tr>
<tr>
<td>Machine</td>
<td>Machine</td>
</tr>
</tbody>
</table>

KB

- NL Generation (Cyc)
- Patterns

- Predefined options (menus)
- Knowledge guided input
- Free text input
## Semantic Chat: Social Conversational Crowdsourcing

<table>
<thead>
<tr>
<th>Questions</th>
<th>Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>Machine</td>
</tr>
<tr>
<td>Machine</td>
<td>Human</td>
</tr>
<tr>
<td>Machine</td>
<td>Machine</td>
</tr>
</tbody>
</table>

- NL Generation (Cyc)
- Patterns
- Predefined options (menus)
- Knowledge guided input
- Free text input
Levels of Knowledge Acqu.

- **Acquisitions of instances**
  - Person: Luka, Dane, Janez

- **Acquisitions of classes**
  - SlovenianPerson -> Person

- **Acquisition of relations**
  - [Person] ResidesInRegion [Region]

- **Acquisition of reasoning rules**
Levels of Knowledge Acqu.

- **Acquisitions of instances**
  - Person: Luka, Dane, Janez  
  
  **Easy**
  - CC, Oren Etzioni et al. (TextRunner), Taylor M. et al. …

- **Acquisitions of classes**
  - SlovenianPerson -> Person
  
  **Trickier**
  - CC, Huairen Lin et al., Di Iorio et al., Mitchell, T. et al., …

- **Acquisition of relations**
  - [Person] ResidesInRegion [Region]
  
  **Hard to do it right**
  - Oren Etzioni et al., Luis von Ahn et al., Eckert, K. et al., Mitchell, T. et al.,

- **Acquisition of reasoning rules**
  
  **Hard**
  - ?
Instances

• Relation: [Person] likesAsFriend [Person]
  – Arg1Isa: Person
  – Arg2Isa: Person
  – Pattern: “[Person1] and [Person2] are friends.”

• Example: Luka and ___ are friends.
  – User fills the pattern with i.e. „Blaž“
  – System implies: (isa Blaž Person)
Instances

- Relation: [Person] likesAsFriend [Person]
  - Arg1Isa: Person
  - Arg2Isa: Person
  - Pattern: "[Person1] and [Person2] are friends."

- Example:
  - User fills the pattern with i.e. "Blaž"
  - System implies: (isa Blaž Person)

I don’t seem to have "Marko Grobelnik" before. What kind of thing is it?

Back

[individual] employs Luka Bradeško.

Big Boss

I don’t seem to have "Big Boss" before. What kind of thing is it?

individual Back
Classes

- Relation: [Restaurant] restaurantHasMenuItem [FoodOrDrink]
  - Arg1Isa: Restaurant
  - Arg2Genl: FoodOrDrink
  - Pattern: “[Restaurant] has [FoodOrDrink] on the menu.”

- Example: Egoist has ___ on the menu.
  - Answer: Coffee with milk
  - System implies: (subclass CoffeWithMilk FoodOrDrink)
Classes

- Relation: [Restaurant] hasMenuItem
  - Arg1Isa: Restaurant
  - Arg2Genl: FoodOrDrink
  - Pattern: "[Restaurant] has [FoodOrDrink] on the menu."

- Example: Egoist has ___ on the menu.
  - Answer: Coffee with milk
  - System implies: (subclass CoffeWithMilk FoodOrDrink)

- Example: Luka Bradeško had a coffee with milk at his or her 15:11:40, September 14, 2012 visit to Egoist Lounge Bar.

- Example: Luka Bradeško had [food and drink] at his or her 19:07:18, September 18, 2012 visit to Egoist Lounge Bar.

- Example: Janez: Janez Starc had a coffee with milk at Janez

FoodOrDrink
  - Coffee
  - Alcoholic Drink
    - Cappuccino
    - Beer
Classes

- Default position is not accurate enough.
- Better positioning with asking more questions
  - Is Coffee with milk type of Drink?
  - Is Coffee with milk type of Coffee?
- Better positioning based on string presentation.
  - Because of word “coffee” in Coffee with milk we can ask directly whether it is a type of Coffee.
Classes

• Default position is not accurate enough.

• Better positioning with asking more questions:
  – Is Coffee with milk type of Drink?
  – Is Coffee with milk type of Coffee?

• Better positioning based on string presentation:
  – Because of word “coffee” in “Coffee with milk” we can ask directly whether it is a type of Coffee or Drink.

Luka Bradeško had [food and drink] at his or her 19:07:18, September 18, 2012 visit to Egoist Lounge Bar.

Stranger Coffee

I don’t seem to have “strange coffee” before. What kind of thing is it?

coffee  food and drink  type of

Janet: Janet Starc had a coffee with milk at Janet

0 5 0 2 3  Egoist Lounge Bar
Relations

• We don’t do it via ontology supported KA yet
• It is possible to find new relations on grounded ontologies
  – Klemen Simonič et. al.: Predicting Instance Properties in Linked Data
• Fugure out new relations based on user to user questions/answers
  – Janez Starc: Identifying good patterns for relation extraction
Rules

- For now used to produce new questions and acquiring new knowledge.
- Rule Example:
  - \((\text{implies}\ (\text{levelOfSecurity}\ (\text{WirelessFn}\ ?X)\ \text{PasswordProtected})\ (\text{generateSentenceForTerm}\ ?X)\ (\text{passwordForWiFi}\ (\text{WirelessFn}\ ?X)\ ?Y)))\)

- We leave rule acquisition for the future research
Ontology driven Knowledge Acquisition Challenges

- Placement of the knowledge into existing KB
- Acquisition of Relations
- Acquisition of Rules
- Generation of patterns is almost as expensive as building ontology itself
Crowdsourcing

- The Knowledge Acquisition Forms are presented to users based on their context
- Some answers get stored in private parts of KB due to privacy
- Public answers are presented to other users to confirm or deny them
- Massively denied answers are removed and user producing them marked as less relevant
Crowdsourcing

• The Knowledge Acquisition Forms are presented to users based on their context.
• Some answers are stored in private parts of KB due to privacy.
• Public answers are presented to other users to confirm or deny.
• Massively denied answers are removed and user producing them marked as less relevant.
Crowdsourcing Challenges

• Quality/truth/consistency control
  – Different truths for different users

• Maintaining user involvement
  a) Pay for users (Mechanical Turk)
  b) Other ways of motivating (i.e. wrap into some other activity)
    - Order and appeal of the questions is important
  - Maximizing KA usefulness of the questions
## Evolution of Conversational Clients

<table>
<thead>
<tr>
<th>Year</th>
<th>Chatbot</th>
<th>Technology</th>
<th>Language Tricks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1966</td>
<td>Eliza</td>
<td>Simple pattern matching</td>
<td></td>
</tr>
<tr>
<td>1991</td>
<td>PC Therapist III (Weintraub, 1986)*</td>
<td>parsing, pattern matching, word vocabulary, remembers sentences</td>
<td>non sequitur, canned responses</td>
</tr>
<tr>
<td>1992</td>
<td>PC Professor (Weintraub, 1986)*</td>
<td>Pattern matching, database like system</td>
<td>Model of personal history</td>
</tr>
<tr>
<td>1993</td>
<td>PC Politician (Weintraub, 1986)*</td>
<td>Pattern matching, Markov chain models to construct some replies</td>
<td>database of trick sentences, Model of personal history, not repeating itself</td>
</tr>
<tr>
<td>1994</td>
<td>TIPS (Whalen, 1994; Hutchens, 1997)</td>
<td>Pattern matching, database like system</td>
<td></td>
</tr>
<tr>
<td>1995</td>
<td>PC Therapist (Weintraub, 1986)*</td>
<td>Same as in 1991</td>
<td>Same as in 1991</td>
</tr>
<tr>
<td>1996</td>
<td>HeX (Hutchens, 1997)</td>
<td>Statistical parser, pattern matching, modular with weighted modules, WordNet synonyms, list of proper names, ontology, database for storing facts</td>
<td>Proactivity</td>
</tr>
<tr>
<td>1997</td>
<td>CONVERSE (Levy, 1997)</td>
<td>Pattern matching, database like system</td>
<td></td>
</tr>
<tr>
<td>1998</td>
<td>Albert One (Garner, 1995)</td>
<td>Pattern matching, hierarchical composition of other chatbots (Eliza, Fred, Sextalk)</td>
<td>Proactive monologues</td>
</tr>
<tr>
<td>1999</td>
<td>Albert One (Garner, 1995)</td>
<td>Pattern matching, hierarchical composition of other chatbots (Eliza, Fred, Sextalk)</td>
<td>Proactive monologues</td>
</tr>
<tr>
<td>2002</td>
<td>Ella (Copple, 2009)</td>
<td>Parser, pattern matching – simpler than AIML, Markov Chains, Context free grammar (CFG)</td>
<td>Monologues, not repeating itself; identify gibberish, play knock-knock jokes</td>
</tr>
<tr>
<td>2003</td>
<td>Jabberwock (Pirner, 2003)</td>
<td>Same as in 2000</td>
<td></td>
</tr>
<tr>
<td>2006</td>
<td>Joan (Carpenter, 2006)</td>
<td>Same as in 2000</td>
<td></td>
</tr>
<tr>
<td>2007</td>
<td>UltraHAL by Robert Medeksza*</td>
<td>Combination of VB code and pattern matching scripts</td>
<td></td>
</tr>
<tr>
<td>2008</td>
<td>Elbot (Roberts, 2007)*</td>
<td>Commercial NLI system</td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td>Do-Much-More (Levy, 2009)*</td>
<td>Commercial property of Intelligent Toys Ltd.</td>
<td></td>
</tr>
</tbody>
</table>
Conversational Clients

• Mostly built to pass the Turing Test (Loebner prize competition)
  – Lot’s of pretending. Some of these go so far to fake spelling mistakes.
  – Mostly just lots of patterns. Quality depends on number of hard coded patterns.

• No REAL breakthrough since Eliza
  – With exception of totally different approach in Cleverbot (2005,2006). But this one is impossible to control or influence, so it seems it is dead end.
Conversational Clients

• Example, 2 Cleverbots talking to each other:
  – [http://www.youtube.com/watch?v=WnzIbyTZsQY](http://www.youtube.com/watch?v=WnzIbyTZsQY)
XML Based patterns:

```xml
<category>
  <pattern> I NEED HELP * </pattern>
  <template>Can you ask for help in the form of a question? </template>
</category>
```

Allows Recursion:

=> Can you please tell me what LINUX is right now?
<category>
  <pattern> * RIGHT NOW </pattern>
  <template> <srai><star/></srai></template>
</category>

=> CAN YOU PLEASE TELL ME WHAT LINUX IS and then
<category>
  <pattern> CAN YOU PLEASE * </pattern>
  <template> <srai> Please <star/></srai></template>
</category>
ChatScript

- AIML’s successor.
  - Better syntax to make it easier to maintain.
  - Fixes zero word matching problems
  - Introduces
    - Continuations (followups)
    - Logical and/or variables in patterns
    - Facts (triplets)
    - Variables
    - functions
ChatScript

Simpler patterns + concepts:

\textbf{concept:} \textit{~meat} (bacon ham beef meat flesh veal lamb chicken pork steak cow pig)

\texttt{s: (I love ~meat) Do you really? I am a vegan.}

Continuations:

\texttt{s: (I like spinach) Are you a fan of the Popeye cartoons?}
\hspace{1cm} \texttt{a: (yes) I used to watch him as a child. Did you lust after Olive Oyl?}
\hspace{2cm} \texttt{b: (no) Me neither. She was too skinny.}
\hspace{3cm} \texttt{b: (yes) You probably like skinny models.}
\hspace{1cm} \texttt{a: (no) What cartoons do you watch?}
\hspace{2cm} \texttt{b: (none) You lead a deprived life.}
\hspace{3cm} \texttt{b: (Mickey Mouse) The Disney icon.}
Conversational Clients Challenges

• Maintaining the context of the conversation
• Remembering the facts
• Chinese room problem (not actually understanding the conversation)
Conclusions on our approach

• Improves knowledge acquisition regarding the precision
  • Consistency control
  • Controlled entry of facts
  • Bad entries gets outvoted
  • Does KA as a side effect of using AI

• Rigid in terms of conversation
  – Only one way of saying things
  – Questions and answers are not prioritized
  – Variability is in terms of intelligence and not in terms of conversation (the opposite problem of Chat Clients)
Next Steps

• On top of some technical issues:
  – Combine the Chat Client approach to the conversation with the Ontology driven knowledge acquisition
  – Find methods to weight facts and questions by interestingness to the user
    • Collect the data about how and when are users answering/skipping particular question
  – Collect the user to user questions to be able to infer new relations and patterns
Questions

• luka.bradesko@ijs.si