Question Answering and the development of the *Hajen* System

**Pierre Nugues**
Joint work with Marcus Klang, Rebecka Weegar, Peter Exner, Juri Pyykkö
Background: IBM Watson

- IBM Watson: A system that can answer questions better than any human
- Video: https://www.youtube.com/watch?v=WFR3lOm_xhE
- Bottom line: Text is the repository of human knowledge
**Goals of *Hajen***

- Build a system that can answer questions in a way similar to IBM Watson
- Beyond Swedish:
  - Implement a generic, open-source, modular, question answering platform
  - Make it easy to integrate multilingual components
  - Use of full-text indexing, graph-based document models, map-reduce Hadoop ecosystem, dependency injection techniques
Hajen?

- Part of Vetenskapsrådets frame funding Det digitaliserade samhället
- Semantic processing project: Mot kunskapsbaserad storskalig kunskapsutvinning ur svenskt text
- Cooperation with Gothenburg University and Chalmers
- The name is a tribute to Ulf Hannerz, the legendary winner of Kvitt eller dubbelt in 1957.
- http://www.svtplay.se/klipp/297574/10-000-kronorsfragan-kvitt-eller-dubbel
Question-Answering Architecture
(simplified from IBM Watson)

Question processing

Passage retrieval

Answer extraction

Question parsing and classification:
Syntactic parsing, entity recognition, answer classification

Document retrieval. Extraction and ranking of passages:
Indexing, vector space model.

Extraction and ranking of answers:
Answer parsing, entity recognition
Corpus: *Kvitt eller dubbelt* (*KED*)

- In language processing, everything starts with a corpus
- *SVT-klassikern*, a simplified *Jeopardy!* in Sweden
Kvitt eller dubbelt: The Questions
Representing the Questions

- We transcribed the 2300 questions and we represented them as an RDF graph.
- A question from the category: “Får jag lov”

Q: Vad heter den argentinska dans som Petra Nielsen dansade i melodifestivalen 2004? A: Tango

- Represented as:

  [kvitt:line 1; kvitt:value 250; kvitt:text "Vad heter den argentinska dans som Petra Nielsen dansade i melodifestivalen 2004?"; kvitt:answer "Tango"]

- Using SPARQL, we can extract easily data from the graph and carry out tests.
Passage Retrieval: Textual Resources

- Question answering needs a knowledge source in the form of a collection of documents.
- Wikipedia has a large and growing coverage of topics.
- It is easy to download from dumps.wikimedia.org.
- Is the Wikipedia suitable to answer *Kvitt eller dubbelt* questions?
Passage Retrieval: Indexing

• We segmented the Wikipedia articles into paragraphs: the passages.
• We indexed the passages using Apache Lucene
• Given a question, like:

  Vad heter den argentinska dans som Petra Nielsen dansade i melodifestivalen 2004?

  We return the most relevant paragraphs scored using the Lucene’s built-in TF.IDF measure.

• How good is it?
Passage Retrieval: Some Results

We submitted questions from the corpus and we checked if the returned paragraphs contained the answer string.
Answer Extraction

• Most KED answers are concepts or entities (real or imaginary unique things).

• Corresponds to common nouns or proper nouns in the passages.

• Extractible using a part-of-speech tagger.

• Proper nouns can be classified into categories such as persons, cities, countries, organizations, etc.

• Once extracted, candidates are ranked by increasing frequency.

• This is our baseline answer extraction
<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Petra</td>
<td>Petra</td>
<td>PM</td>
<td>(person</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Magdalena</td>
<td>Magdalena</td>
<td>PM</td>
<td>person</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Nielsen</td>
<td>Nielsen</td>
<td>PM</td>
<td>person</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>,</td>
<td>,</td>
<td>MID</td>
<td>_</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ursprüngligen</td>
<td>ursprüngligen</td>
<td>AB</td>
<td>_</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Inczèdy-Gombos</td>
<td>inczèdy-gombos</td>
<td>PM</td>
<td>_</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>,</td>
<td>,</td>
<td>MID</td>
<td>_</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>född</td>
<td>föda</td>
<td>PC</td>
<td>_</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1RG</td>
<td>_</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>februari</td>
<td>februari</td>
<td>NN</td>
<td>_</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1965</td>
<td>1965</td>
<td>RG</td>
<td>_</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>i</td>
<td>i</td>
<td>PP</td>
<td>_</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Stockholm</td>
<td>Stockholm</td>
<td>PM</td>
<td>(place)</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>.</td>
<td>.</td>
<td>MAD</td>
<td>_</td>
<td></td>
</tr>
</tbody>
</table>
### Part-of-Speech Tagging

<table>
<thead>
<tr>
<th></th>
<th>Name</th>
<th>Form</th>
<th>Tag</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Petra</td>
<td>Petra</td>
<td>PM</td>
<td>(person)</td>
</tr>
<tr>
<td>2</td>
<td>Magdalena</td>
<td>Magdalena</td>
<td>PM</td>
<td>person</td>
</tr>
<tr>
<td>3</td>
<td>Nielsen</td>
<td>Nielsen</td>
<td>PM</td>
<td>person</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>,</td>
<td>MID</td>
<td>_</td>
</tr>
<tr>
<td>5</td>
<td>ursprunngligen</td>
<td>ursprunngligen</td>
<td>AB</td>
<td>_</td>
</tr>
<tr>
<td>6</td>
<td>Inczèdy-Gombos</td>
<td>inczèdy-gombos</td>
<td>PM</td>
<td>_</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>,</td>
<td>MID</td>
<td>_</td>
</tr>
<tr>
<td>8</td>
<td>född</td>
<td>föda</td>
<td>PC</td>
<td>_</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1RG</td>
<td>_</td>
</tr>
<tr>
<td>10</td>
<td>februari</td>
<td>februari</td>
<td>NN</td>
<td>_</td>
</tr>
<tr>
<td>11</td>
<td>1965</td>
<td>1965</td>
<td>RG</td>
<td>_</td>
</tr>
<tr>
<td>12</td>
<td>i</td>
<td>i</td>
<td>PP</td>
<td>_</td>
</tr>
<tr>
<td>13</td>
<td>Stockholm</td>
<td>Stockholm</td>
<td>PM</td>
<td>(place)</td>
</tr>
<tr>
<td>14</td>
<td>.</td>
<td>.</td>
<td>MAD</td>
<td>_</td>
</tr>
</tbody>
</table>
Answer Extraction: Preliminary Results

Results limited to answers classified as entities.
Joint work with Juri Pyykkö and Rebecka Weegar.
The Mark-1 Question-Answering Engine

Answer Type Classification

• From the question text, we can often guess the type of the answer and discard unlikely candidates.

• Q: Vilket är det enda nordiska land man inte kan köra bil från Sverige? A: Island

• We annotated a part of the answers from KED and we trained an answer type classifier.

• We use logistic regression and we get probabilities for the answer types.

• Not completely integrated yet.

Work started with Christopher Käck and Robin Leussier
Reranking

- A reranker is a module that takes a list of ranked candidates and reorder them using “global” features such as the answer type.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Answer</th>
<th>F/T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Europa</td>
<td>false</td>
</tr>
<tr>
<td>2</td>
<td>Patrik Budda Andersson</td>
<td>false</td>
</tr>
<tr>
<td>3</td>
<td>Sannas</td>
<td>false</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>Island</td>
<td>true</td>
</tr>
</tbody>
</table>

Q: *Vilket är det enda nordiska land man inte kan köra bil från Sverige?*

- Using the corpus, it is possible to train a binary classifier with features such as the predicted answer type, word type

- Improves considerably the results (Not yet in the Mark-X line)

Work in cooperation with Jakob Grundström
Structured Data and Inference (From IBM Watson)

WAR MOVIES: A 1902 Joseph Conrad work set in Africa inspired this director to create a controversial 1979 war film.
Named Entity Processing

- A first step to inference is to identify and disambiguate named entities.
- Two techniques: named entity recognition (NER) and named entity disambiguation (NED)

  Göran Persson var statsminister mellan åren 1996 och 2006 samt var partiledare för Socialdemokraterna

- Google’s motto Things not Strings
Named Entity Recognition

- Named entity recognition (NER) tags the words that correspond to named entities, possibly with their type.
- Common types include person, organization and location.
  - Göran Persson is a person.
  - Socialdemokraterna is an organization.
- Techniques to carry out named entity recognition use extensions of part-of-speech tagging.
Named Entities are Ambiguous

Göran Persson kan syfta på

- Göran Persson (född 1960), socialdemokratisk politiker från Skåne
- Göran Persson (musiker), svensk proggmusiker
- Jöran Persson, svensk ämbetsman på 1500-talet
Named Entity Disambiguation

Named entity disambiguation (NED) links words to entities: *strings to things.*

**Göran Persson** var statsminister mellan åren 1996 och 2006 samt var partiledare för **Socialdemokraterna.**
Wikipedia and Wikidata

- Each page in Wikipedia (concepts and entities) has a unique identification number across all the languages, a sort of personnummer.

- The Q-number repository (wikidata) is a growing source of structured data.
  - Göran Persson (statsminister): Q53747
  - Göran Persson (riksdagsledamot): Q5626648
### Wikidata Content

#### In other languages

<table>
<thead>
<tr>
<th>Language</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>français</strong></td>
<td>Gôran Persson, homme politique suédois</td>
</tr>
<tr>
<td><strong>suomi</strong></td>
<td>Gôran Persson</td>
</tr>
</tbody>
</table>

#### Statements

<table>
<thead>
<tr>
<th>Type</th>
<th>Value</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>sex or gender</strong></td>
<td>male</td>
<td>3 references</td>
</tr>
<tr>
<td><strong>LCNAF identifier</strong></td>
<td>n980833850</td>
<td>1 reference</td>
</tr>
<tr>
<td><strong>instance of</strong></td>
<td>human</td>
<td>1 reference</td>
</tr>
</tbody>
</table>
NEDforia: A Disambiguation Tool

Named Entity Disambiguation Result

Göran Persson var statsminister mellan åren 1996 och 2006 samt var partiledare för Socialdemokraterna.

<table>
<thead>
<tr>
<th>Rank</th>
<th>URI</th>
<th>Rank-value</th>
<th>Commonness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>wikidata:Q53747</td>
<td>0.9999942697886333</td>
<td>0.9809523809523801</td>
</tr>
<tr>
<td>2.</td>
<td>wikidata:Q6042900</td>
<td>0.753557930607055</td>
<td>0.0126984126984121</td>
</tr>
<tr>
<td>3.</td>
<td>wikidata:Q6043257</td>
<td>0.7290883542424511</td>
<td>0.0031746031746038</td>
</tr>
<tr>
<td>4.</td>
<td>wikidata:Q2625684</td>
<td>0.7246375610550542</td>
<td>0.0031746031746038</td>
</tr>
<tr>
<td>5.</td>
<td>wikidata:Q5626648</td>
<td>0.6595001745138378</td>
<td>0</td>
</tr>
</tbody>
</table>

Nedforia Marcus Klang 2014

Demonstration: http://semantica.cs.lth.se:8888/sv/ned/query
Mark 1 Architecture

Webforia (HTTP frontend)

Nlpforia

Language (Dependency Injector)

Parsers – Wikimarkup, ConLL

Disambiguation – Detectors and disambiguators

Data – Fulltext, Passage search, Key/Value storage

Classification – Short text (based on LibLinear)

QA – Rankers, Extractors, Hypothesizers, Combiners

NLP Tools – Tokenizers, NER, Segmenters, POS Taggers

Global Dependency Injection

Document Model (Graph based)

Toolforia (CLI)
Mark 1: Features

• **Document model**
  - Directed property graph
  - Built-in small query engine
  - Extends or re-uses ideas from KOSHIK (from Peter Exner)

• **Multilingual and modular**
  - **Idea:** A Language is nothing more than an assemblage of language specific components with models.
  - Constructor based dependency injection (Guice)

• **Designed for prototyping and experimentation**

• **Open source**
Auto annotation was based on Stagger and Maltparser with respective models trained on the Stockholm-Umeå Corpus (SUC)
NEDforia: Disambiguation Results

- Small test corpus consisting of 10 news articles from mixed sources: Aftonbladet, Svd, DN, Nyteknik, HD, IDG and Dagens Industri
- We extracted and annotated 201 named entities, preliminary results.

<table>
<thead>
<tr>
<th>Recall</th>
<th>Precision</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.696</td>
<td>0.748</td>
<td>0.721</td>
</tr>
</tbody>
</table>
Cross-Language Extraction

<table>
<thead>
<tr>
<th>English frame</th>
<th>Swedish frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBJ</td>
<td>SS</td>
</tr>
<tr>
<td>VERB</td>
<td>VERB</td>
</tr>
<tr>
<td>NMOD</td>
<td>NMOD</td>
</tr>
<tr>
<td>NMOD</td>
<td>DT</td>
</tr>
<tr>
<td>OBJ</td>
<td>OO</td>
</tr>
<tr>
<td>TMP</td>
<td>ET</td>
</tr>
<tr>
<td>PMOD</td>
<td>HD</td>
</tr>
</tbody>
</table>

Joint work with Peter Exner
Koshik

• Koshik: End-to-end framework to process multilingual documents
• Based on Hadoop and our Crafoord cluster
• Koshik is a talking elephant (Korean)
Koshik Architecture

- Koshik uses the Avro binary format to serialize the documents.
- Avro is designed for Hadoop and allows other data warehousing tools to directly query the documents.
- Koshik can be queried directly through Hive, which offers an SQL-like query language called HiveQL.
- Available at https://github.com/peterexner/KOSHIK
Querying with HIVE

Number of articles:
> SELECT count(identifier) from koshikdocs;
Job 0: Map: 920  Reduce: 1  Cumulative CPU: 35243.42 sec  HDFS Read: 248123756147
HDFS Write: 8 SUCCESS
OK
4012291

Number of tokens:
> SELECT count(ann) FROM koshikdocs LATERAL VIEW explode(annotations.layer)
annTable as ann WHERE ann LIKE '%Token';
Job 0: Map: 920  Reduce: 1  Cumulative CPU: 35199.98 sec  HDFS Read: 248123756147
HDFS Write: 11 SUCCESS
OK
1485951256

Number of nouns:
> SELECT count(key) FROM (SELECT explode(ann) AS (key,value)
FROM koshikdocs LATERAL VIEW explode(annotations.features) annTable as ann) annmap
decmap WHERE key='PPOS' AND value LIKE 'NN%';
Job 0: Map: 920  Reduce: 1  Cumulative CPU: 46694.33 sec  HDFS Read: 248123756147
HDFS Write: 10 SUCCESS
OK
476161210
A Snapshot of the Future

When wireless is perfectly applied, the whole earth will be converted into a huge brain, which in fact it is, all things being particles of a real and rhythmic whole. We shall be able to communicate with one another instantly, irrespective of distance. Not only this, but through television and telephony we shall see and hear one another as perfectly as though we were face to face, despite intervening distances of thousands of miles; and the instruments through which we shall be able to do this will be amazingly simple compared with our present telephone. A man will be able to carry one in his vest pocket.

Nikola Tesla, *Colliers*, January 30, 1926