Learning By Reading:

Automatic Knowledge Extraction through Semantic Analysis

PhD Proposal • Jesse English • 3/13/2008
Table of Contents

- Motivation
- Proposal
- Requirements
- Results
- Evaluation
- Future Work
Motivation

- Motivation
  - Overview
  - How do we arrive at semantically annotated text?
  - Dodging the bottleneck…
  - Addressing the bottleneck…

- Proposal
- Requirements
- Results
- Evaluation
- Future Work
Motivation: Overview

- Semantically annotated text (natural language text marked up in a machine readable format) has a variety of uses:
  - Opinion extraction (crawling the blogosphere)
  - Topic gisting (summarization and searching)
  - Question answering (alternate search engines)
Motivation: How do we arrive at semantically annotated text?

- By hand?
  - Extremely time consuming
  - Unpredictably error prone (people make mistakes, predicting which ones is difficult)
- Using Natural Language Processing (NLP)
  - Extraordinarily complicated system to produce
  - Needs vast amounts of world knowledge (in the form of a lexicon and ontology)
    - “Knowledge Acquisition Bottleneck”
Motivation: Dodging the bottleneck...

- Automating knowledge acquisition:
  - Structural semantic interconnections [1]
    - “business plan” from “business” and “plan”
  - ML methods over syntactic parse trees [2], [3], [4]
- There is a drawback! These methods are missing semantic information!

1. [Navigli et al. 2004]
2. [Yangarber, 2003]
3. [Reinberger and Spyns, 2004]
4. [Toutanova et al. 2005]
“The man listened carefully to the address, and later was able to find his way there easily.”

- Using a syntactic parse only, one would have to guess the meaning of “address”
- Applying a statistical count, a system would likely see the meaning as that of “a speech”, not “a location”
  - This is due to the position of “address” in the sentence
  - A semantic parse would pick up on this distinction, and would see how “address” is referenced later
Motivation: Addressing the bottleneck...

• The bottleneck is a Catch-22!
  • A good semantic parse cannot be produced without broad coverage…
  • But you can’t get broad coverage without a good semantic parse!

• In order to avoid this, you must have a bootstrapped system to start with
  • A system with a “critical mass” of knowledge, enough to get the ball rolling and keep it rolling as it gains ground!
Proposal

- Motivation
- Proposal
  - Overview
  - Lifetime learning…
  - Selecting a corpus for lifetime learning…
  - The wonders of the world wide web :) 
  - The wickedness of the world wide web :( 
  - Semantic annotation of the text…
  - Constructing candidate knowledge…
  - Broaden the system’s coverage!
- Requirements
- Results
- Evaluation
- Future Work
Proposal: Overview

- Combining NLP and ML to produce a “lifetime learner”
- An NLP system that enhances itself, escaping the acquisition bottleneck
Proposal: Lifetime learning…

- Given an unknown word, scan a corpus for text containing it
- Semantically analyze the text, relaxing on unknowns
- Combine relevant output from the analysis into candidate knowledge
- Add the candidate to the existing knowledge (thus broadening coverage)
Proposal: Selecting a corpus for lifetime learning...

- Any closed corpus (regardless of size) is finite, and therefore cannot provide true lifetime learning.
- The web, however, provides an endless source of material including:
  - Source text
  - Statistical information
- See [Kilgarriff and Grefenstette, 2003]
Proposal: The wonders of the world wide web :)  

- A perfect choice for the system proposed:
  - Endless, domain independent knowledge
    - Domain specific text may require more intimate knowledge about the domain, bringing us back to the Catch-22
  - Written in natural language
  - Easily queried
Proposal: The wickedness of the world wide web :(

- Noise!
  - Erroneous data
    - “fish have four feet”
  - Malformed data
    - This HTML file is actually some encrypted PDF?!?
  - Poorly structured text
    - “bbl, i g2g to th estore 4 a bit!!1”

- Misinterpreted queries!
  - Incorrect keywords
  - Bad indexing
Proposal: Semantic annotation of the text…

- Automatic annotation of the text produces a machine readable semantic parse
- As unknown input is expected (by definition), methods of “relaxation” will need to be used
  - Unidirectional selectional restrictions

The baker baked the XYZ.

baker ⇒ *agent-of* ⇒ bake ⇒ *theme* ⇒ pastry
Proposal: Constructing candidate knowledge...

- Extracting the knowledge from semantic annotations we can create new knowledge for the NLP system
  - The knowledge should be filtered
  - The knowledge should also be clustered (words tend to be polysemous, so deciding how many senses there are, and what learned knowledge belongs to which is important)
  - Restructure the learned knowledge into world knowledge for the NLP system
Proposal: Broaden the system’s coverage!

- Append the new knowledge to the existing knowledge
  - Depending on the way the knowledge is organized (hierarchically for example, as in an ontology) this must be done carefully
  - After this is done, assuming the knowledge added is accurate, the system’s coverage has been broadened
    - Increasing it’s use in other applications, in addition to it’s ability to continue learning
Requirements

- Motivation
- Proposal
- Requirements
  - Presupposed existing systems...
  - Google
  - OntoSem
  - DEKADE
  - WEKA
  - others
- Results
- Evaluation
- Future Work
Requirements: Presupposed existing systems…

- Access to an open corpus
- A natural language processing system
- An interactive environment into the NLP system
- Machine learning tools
- Various low-level (implementation only) tools
  - Databases
  - HTML parsers
Requirements: Existing systems (Google)...

- To gain query access to the web, and simultaneously gain access to statistical data (such as page hit counts), Google (and it’s freely available SOAP Search API) is a perfect fit
  - Indexed web pages can be returned based on a series of search parameters
  - Minor word processing is done by Google to broaden search results (such as root word processing and searching)
Requirements: Existing systems (OntoSem)...

- To fill the need for a natural language processor, OntoSem fits the bill
  - A fully automatic text processing system
  - Relaxes constraints (uses unidirectional selectional restrictions)
  - Is dependent on the quality and coverage of its static knowledge
  - Produces output in a similar format to its static knowledge input
Requirements: Existing systems (DEKADE)

- To fully utilize and explore OntoSem, its knowledge, and the output it produces, an interface to the system (both user, and programmer level) is needed
  - DekadeAPI
  - DekadeAtHome
Requirements: Existing systems (WEKA)

- To make full use of the latest ML tools, (specifically clustering algorithms), the WEKA toolkit provides the perfect platform
  - EM algorithm
Requirements: Existing systems (others)

- PostgreSQL (http://www.postgresql.org/)
- HTML Parser (http://htmlparser.sourceforge.net/)
Results

- Motivation
- Proposal
- Requirements
- Results
  - The first experiment…
  - The second experiment…
  - The third experiment…
- Evaluation
- Future Work
• The first experiment, published in AAAI Spring Symposium 2007, consisted of running the process on four words

• The general flow of the experiment was consistent with the process described, with “less sophistication”:
  • Clustering for multiple senses was not done
  • Less filtering of junk was performed
  • Placement in the ontology was done by using the OntoSearch algorithm [Onyshkevych, 1997]. This method has since been shown to be an inaccurate method of ranking for this experiment.
## Results: The first experiment...

<table>
<thead>
<tr>
<th>Word</th>
<th>Best Match</th>
<th>Selected Match</th>
<th>Difference</th>
<th>Rank</th>
<th>Percentile</th>
</tr>
</thead>
<tbody>
<tr>
<td>pundit</td>
<td>TELEVISION, CITIZEN, HUMAN (and 12 more) 0.800</td>
<td>INTELLECTUAL 0.679</td>
<td>0.121</td>
<td>210/~6000</td>
<td>3.5%</td>
</tr>
<tr>
<td>CEO</td>
<td>EVENT 0.900</td>
<td>PRESIDENT-CORPORATION 0.618</td>
<td>0.262</td>
<td>&gt;500/~6000</td>
<td>&gt;8.3%</td>
</tr>
<tr>
<td>hobbit</td>
<td>PUBLISH 0.900</td>
<td>HUMAN 0.806</td>
<td>0.094</td>
<td>18/~6000</td>
<td>0.3%</td>
</tr>
<tr>
<td>song</td>
<td>WORD, RECORD-TEXT, OBJECT (and 8 more) 0.800</td>
<td>SONG 0.800</td>
<td>0.000</td>
<td>12/~6000</td>
<td>0.2%</td>
</tr>
</tbody>
</table>
Results: The first experiment...

- Used a small generated corpus
- Did not consider multiple word senses
- Used an improper ranking algorithm
- Used words whose senses already were found in the lexicon/ontology
Results: The second experiment…

- To improve the first experiment several steps were taken:
  - Implementation of an appropriate ranking algorithm (abandoning OntoSearch)
  - Improved filtering
  - Larger generated corpus
  - Targeting unknown word senses
Results: The second experiment...

<table>
<thead>
<tr>
<th>Word (4 of 12)</th>
<th>Similarity to DINOSAUR</th>
<th>Similarity to best match</th>
<th>Rank (out of ~16913)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brontosaurus</td>
<td>0.373</td>
<td>0.492</td>
<td>9007</td>
</tr>
<tr>
<td>Diplodocus</td>
<td>0.500</td>
<td>0.550</td>
<td>2290</td>
</tr>
<tr>
<td>Stegosaurus</td>
<td>0.499</td>
<td>0.538</td>
<td>625</td>
</tr>
<tr>
<td>Triceratops</td>
<td>0.482</td>
<td>0.488</td>
<td>588</td>
</tr>
</tbody>
</table>
Results: The third experiment...

- The third (and current) experiment involves a few major changes to the process:
  - Multiple word senses are considered
  - Clustering is used to propose word senses
  - A “decision tree” is used as part of the similarity measurement process
  - Substantially larger corpus used (minimum 1000 sentences per target word)
Results: The third experiment...

<table>
<thead>
<tr>
<th>Word</th>
<th># Proposed Clusters</th>
<th>Word</th>
<th># Proposed Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>5</td>
<td>kid</td>
<td>3</td>
</tr>
<tr>
<td>artery</td>
<td>2</td>
<td>library</td>
<td>6</td>
</tr>
<tr>
<td>buoy</td>
<td>5</td>
<td>nail</td>
<td>4</td>
</tr>
<tr>
<td>catalogue</td>
<td>6</td>
<td>present</td>
<td>4</td>
</tr>
<tr>
<td>fork</td>
<td>3</td>
<td>rain</td>
<td>4</td>
</tr>
<tr>
<td>free</td>
<td>3</td>
<td>triangle</td>
<td>7</td>
</tr>
<tr>
<td>heart</td>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Results: The third experiment..

<table>
<thead>
<tr>
<th>Cluster head</th>
<th>Closest match</th>
<th>Match value</th>
</tr>
</thead>
<tbody>
<tr>
<td>THEME-OF UTILIZE</td>
<td>FAMILY TRIBE</td>
<td>0.423</td>
</tr>
<tr>
<td>RELATION TUNE-ARTIFACT</td>
<td>COALITION</td>
<td>0.384</td>
</tr>
<tr>
<td>THEME OBJECT</td>
<td>EXTORTION</td>
<td>0.448</td>
</tr>
</tbody>
</table>

Generated TMR Frames for “fork”

- ATTRIBUTE
- CITY
- EVENT
- FORK
- PLACE
Evaluation

- Motivation
- Proposal
- Requirements
- Results
- Evaluation
  - Per candidate?
  - Spiral method!
- Future Work
Evaluation: Per candidate?

- One method of evaluation is at the per candidate level:
  - Given candidate knowledge (an ontology or lexicon entry), it can be compared to a gold standard human-created version
  - It could also be compared to a pre-existing, “closest approximation” (as in the first experiment)
  - The same candidate could also be evaluated by the amount of work required (by hand) to turn it into a gold standard
Evaluation: Spiral method!

- Create a baseline of TMRs
- Learn some amount of unknown words in those TMRs, add the candidates to the static knowledge, and recreate the TMRs
- Repeat again
- This should produce two deltas (change in TMR qualities from the baseline, to the first learned values, and then to the second)
- This (theoretically) shows how adding knowledge both improves TMRs, and as a consequence, improves the learning process
Future Work

- Motivation
- Proposal
- Requirements
- Results
- Evaluation
- Future Work
  - Phase 1
  - Phase 2
  - Phase 3
Future Work: Phase 1

- Improvement of each step of the process, so that better and better results are passed forward
  - Improved querying
  - Better filters to eliminate junk and noise
  - Improved clustering (or sense distinguishing)
  - Improved comparison between candidates and existing concepts
Future Work: Phase 2

- Implementation of the “spiral method”
  - Select a set of semantically related terms to learn
  - Divide the set into two groups
  - Learn all words
  - Manually correct the first group
  - Add the uncorrected first group to the ontology, and re-learn the second group
  - Add the correct first group to the ontology, and re-learn the second group
  - Compare the three resulting group twos
Future Work: Phase 3

- Using the set of words from Phase 2 as a search query, automatically produce a set of TMRs
  - Add the learned words to the ontology, and re-produce the same set of TMRs
  - Produce the same set of TMRs by hand
  - Judge the quality of the three sets of TMRs (hopefully showing improvement towards the gold standard over the baseline when adding in the learned knowledge)
Conclusion

- Proposed a system that combines NLP and ML to create a self-improving lifetime learner
- Suggested a list of available tools to accomplish such a task
- Provided results from previous experiments using this methodology
- Presented some methods of evaluating the results of such a system
- Laid out a plan for future research
Questions?

[Navigli et al. 2004]

[Yangarber, 2003]

[Reinberger and Spyns, 2004]

[Toutanova et al. 2005]

[Kilgarriff and Grefenstette, 2003]

[Onyshkevyych, 1997]