

Part I

Extraction of Generics

What are Generic Expressions?

- ▶ “Birds fly.”
- ▶ “Students are learning.”
- ▶ “A lion is a dangerous animal.”
- ▶ “Mary smokes after dinner.”

Why are Generic Expressions Interesting?

- ▶ Relatively common
- ▶ Abstract and general knowledge
- ▶ Easy to harvest (hopefully)

How do we extract them?

- ▶ Try the machine learning approach!
- ▶ ACE2005 contains annotated data

Table 2 ACE05 Entity Classes

Type	Description
SPC	A particular, specific and unique real world entity
GEN	A kind or type of entity rather than a specific entity
NEG	A negatively quantified (usually generic) entity
USP	An underspecified entity (e.g., modal/uncertain/...)

What do we have to do?

1. Extract examples for sentences containing generic expressions from the corpus
2. Look at the data, develop ideas for features
3. Extract feature values from the corpus
4. Test them using appropriate classifier
5. Make a comprehensive error analysis
6. Apply the trained model to free text, make a manual evaluation

References

Gregory Norman Carlson. Generic Terms and Generic Sentences. *Journal of Philosophical Logic*, 11(2):145–181, 1982.

Gregory Norman Carlson and Francis Jeffry Pelletier, editors. *The Generic Book*. University of Chicago Press, Chicago, 1995.

The ACE 2005 Evaluation Plan. NIST, 2005. URL
<http://www.itl.nist.gov/iad/mig/tests/ace/ace05/>.

Sangweon Suh. Extracting Generic Statements for the Semantic Web. Master's thesis, University of Edinburgh, 2006.

Part II

RTE using ASP

What is RTE?

- ▶ RTE: Recognizing Textual Entailment
- ▶ Task: Decide, whether the text entails the hypothesis

Example

- ▶ **T** In 1998, the General Assembly of the Nippon Sei Ko Kai (Anglican Church in Japan) voted to accept female priests.
- ▶ **H** The Anglican church in Japan approved the ordination of women.

Example

- ▶ **T** Lyon is actually the gastronomic capital of France.
- ▶ **H** Lyon is the capital of France.

What is ASP?

- ▶ ASP: Answer Set Programming
- ▶ “[ASP] is a kind of logic programming with negation as failure that works by translating the logic program into ground form and then searching for stable models [...] using propositional model checking techniques.”
- ▶ Non-Monotonic
- ▶ Fast
- ▶ Implementations are available

How can ASP help us for RTE?

- ▶ Nutcracker: An RTE system based on DRT and CCG, Entailment core is done by First-order logic reasoners and model builders
- ▶ Integration of external knowledge bases (WordNet, SUMO, ...) causes complexity issues
- ▶ Using fast ASP solvers, we could
 - ▶ Use more and larger external knowledge bases
 - ▶ Benefit from “Unsharp”, e.g., generic knowledge
 - ▶ Learn about ASP :)

What do we have to do?

1. Get familiar with ASP solvers
2. Integrate ASP solver into nutcracker, make the necessary adjustments
3. Implement interfaces to external knowledge base(s)
4. Develop a testing scenario for their contribution
5. Identify cases, where the system could benefit from generic knowledge
6. Evaluate the impact of generic knowledge

References

- Johan Bos and Katja Markert. Recognising Textual Entailment with Robust Logical Inference. In *Proceedings of MLCW*, pages 404–426, 2006a.
- Johan Bos and Katja Markert. When logical inference helps determining textual entailment (and when it doesn't). In *Pascal, Proceedings of the Second Challenge Workshop, Recognizing Textual Entailment*, 2006b.
- Michael Gelfond. Answer sets. In *Handbook of Knowledge Representation*. Elsevier Science, 2007.
- Ian Niles and Adam Pease. Towards a Standard Upper Ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems*, 2001.
- Stuart Russel and Peter Norvig. *Artificial Intelligence - A Modern Approach*. Prentice Hall, 2nd edition, 1995.

Part III

Author Identification

Author Identification

What do we want to do?

- ▶ Can we detect if texts are written by the same author?
- ▶ Can we classify texts w.r.t. to their authors (on a large scale)?

Example (Federalist Papers)

- ▶ 1787-1788: Hamilton, Jay and Madison write 85 short essays in support of the U.S. Constitution
- ▶ Published under pseudonym “Publius”, authorship of 12 papers was in dispute
- ▶ Exact authorship is important, as these papers are a primary source for the interpretation of the U.S. Constitution
- ▶ Mosteller and Wallace (1964) identify all 12 papers to be written by Madison, using frequency of function words

Texts and Features

Corpora

- ▶ Projekt Gutenberg
- ▶ British or American National Corpus
- ▶ Die Zeit online
- ▶ ...

Ideas for Features

- ▶ Shallow: Frequency of words or POS tags as n-grams, punctuation, parentheses, word or sentence length, ...
- ▶ Deep: Length of phrases, “syntactic complexity” of sentences, ...

What do we have to do?

1. Establish a baseline
2. Extract additional (or different) features
3. Make a comprehensive error analysis on different kinds of corpora
4. Go back to 2

References

- Michael Gamon. Linguistic Correlates of Style: Authorship Classification with Deep Linguistic Analysis Features. In *Proceedings of COLING*, 2004.
- Moshe Koppel, Shlomo Argamon, and Anat R. Shimoni. Automatically Categorizing Written Texts by Author Gender. *Literary and Linguistic Computing*, 17(4):401–412, 2003.
- Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.
- Frederick Mosteller and David L. Wallace. *Inference and Disputed Authorship*. Addison-Wesley, 1964.
- Efstathios Stamatatos, Nikos Fakotakis, and George Kokkinakis. Automatic Text Categorization in Terms of Genre and Author. *Computational Linguistics*, 26(4):471–495, 2000.