Open Information Extraction: System ReVerb
Fader et al. (2011) Identifying Relations for Open Information Extraction

• Goal:
  - extract relation phrases, i.e., phrases that denote relations in English
  - consider arbitrary relations

• System: ReVerb
  - Source code and data at dfki/data/ReVerb-2012
  - Uses OpenNLP only for POS tagging and chunking
Observed restrictions of current approaches

- Incoherent extractions:
  - relation phrase has no meaningful interpretation; can happen, because learned extractors only make sequence of decisions
  - solved by defining syntactic constraints: every multiword relation phrase must be of form VERB X* PREP

- Uninformative extractions
  - are extractions that omit critical information
  - solved by requiring relation phrases to be light verb constructions, e.g., "(faust, made a deal with, the devil)" instead of "(faust, made, a deal)"

- Avoid overly-specific relations: a relation phrase must been observed with a minimal number of distinct arguments

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Incoherent Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>The guide contains dead links and omits sites.</td>
<td>contains omits</td>
</tr>
<tr>
<td>The Mark 14 was central to the torpedo scandal of the fleet.</td>
<td>was central torpedo</td>
</tr>
<tr>
<td>They recalled that Nungesser began his career as a precinct leader.</td>
<td>recalled began</td>
</tr>
</tbody>
</table>

Table 1: Examples of incoherent extractions. Incoherent extractions make up approximately 13% of TEXTRUNNER’s output, 15% of WOE pos’s output, and 30% of WOE par se’s output.

| is | is an album by, is the author of, is a city in |
| has | has a population of, has a Ph.D. in, has a cameo in |
| made | made a deal with, made a promise to |
| took | took place in, took control over, took advantage of |
| gave | gave birth to, gave a talk at, gave new meaning to |
| got | got tickets to, got a deal on, got funding from |

Table 2: Examples of uninformative relations (left) and their completions (right). Uninformative relations occur in approximately 4% of WOE par se’s output, 6% of WOE pos’s output, and 7% of TEXTRUNNER’s output.
About Syntactic Constraints on Relation Phrases

- **Purpose:** eliminate incoherent and reduce uninformative relation phrases

- **POS tag pattern**
  - a single verb $V$
  - ... followed by a prep $P$
  - ... followed by sequence of Ws and ends with a $P$

- **Heuristics**
  - prefer longest matches
  - merge adjacent sequences („wants to extend“)
  - relation phrase should appear between two argument NPs

Figure 1: A simple part-of-speech-based regular expression reduces the number of incoherent extractions like *was central torpedo* and covers relations expressed via light verb constructions like *gave a talk at.*
About Lexical Constraints

- Problem: syntactic constraints might match very specific rare idiosyncratic instances, e.g.,

  „The Obama administration is offering only modest greenhouse gas reduction targets at the conference.“ →
  (Obama administration, is offering only modest greenhouse gas reduction targets at, conference)

- Solution: lexical constraint

  • a valid relation phrase should take many different arguments in a large corpus
Limitations

• How much recall is lost? Analyze Wu & Weld’s 300 randomly selected Web sentences

• Manual annotation of all relation phrases
  • 327 phrases from which 85% fulfill constraints

• Errors reveal that more than regexpr complexity - is required, e.g., dependency parsing
  - not suitable for Web scale!

NOTE: for English only!

<table>
<thead>
<tr>
<th>Binary Verbal Relation Phrases</th>
</tr>
</thead>
<tbody>
<tr>
<td>85%</td>
</tr>
<tr>
<td>Satisfy Constraints</td>
</tr>
<tr>
<td>8%</td>
</tr>
<tr>
<td>Non-Contiguous Phrase Structure</td>
</tr>
<tr>
<td>Coordination: X is produced and maintained by Y</td>
</tr>
<tr>
<td>Multiple Args: X was founded in 1995 by Y</td>
</tr>
<tr>
<td>Phrasal Verbs: X turned Y off</td>
</tr>
<tr>
<td>4%</td>
</tr>
<tr>
<td>Relation Phrase Not Between Arguments</td>
</tr>
<tr>
<td>Intro. Phrases: Discovered by Y, X . . .</td>
</tr>
<tr>
<td>Relative Clauses: . . . the Y that X discovered</td>
</tr>
<tr>
<td>3%</td>
</tr>
<tr>
<td>Do Not Match POS Pattern</td>
</tr>
<tr>
<td>Interrupting Modifiers: X has a lot of faith in Y</td>
</tr>
<tr>
<td>Infinitives: X to attack Y</td>
</tr>
</tbody>
</table>

Table 3: Approximately 85% of the binary verbal relation phrases in a sample of Web sentences satisfy our constraints.
ReVerb - relation-driven extraction

- **Steps**
  - identification of relation phrases for relation part
  - selection of noun chunks for argument part
  - assigning weights to extracted relation using a logistic regression classifier

- **Novelties**
  - relation phrase is identified “holistically“ and not word-by-word
  - potential phrases are filtered by corpus statistics
  - „relation first“ approach instead of arguments first, which enable better binding of nouns as modification of relation phrases
Extraction Algorithm

- Input: POS-ed and NP-chunked sentence, Output: a set of \((x, r, y)\)

- For each sentence \(s\) do:
  
  - **relation extraction**: for each verb \(v\) in \(s\), find the longest sequence of words \(r_v\), s.t.,
    
    - (1) \(r_v\) starts at \(v\), (2) \(r_v\) satisfies the syntactic constraints, (3) \(r_v\) satisfies the lexical constraints.
    
    - if any pair of matches are adjacent or overlap, then merge them

  - **argument extraction**: for each extracted relation \(r\) do
    
    - left argument: find nearest NP chunk \(x\) to the left of \(r\) that is not a relative pronoun, WHO-adverb or existential „there“
    
    - right argument: find nearest noun phrase \(y\) to the right of \(r\) in \(s\).

- return found \((x, r, y)\) as relation
Validating Lexical Constraints

- To check whether $r_v$ is valid, use a large dictionary of $D$ relation phrases that are known to take many arguments:

  - $D$ is constructed by applying the patterns in a corpus of 500 million web sentences.

  - Set $D$ to be the set of all relation phrases that take at least $k$ distinct argument pairs in the set of extraction.

  - Normalize relation phrases: remove inflection, auxiliary verbs, adjectives, adverbs.

  - Sample test shows $k=20$ as a good value, which results in a set of approx 1.7 million relation phrases stored in memory for extraction time.
Confidence Function

• Goal: Trade recall for precision

• Use logistic regression classifier to assign a confidence score to each extraction

• Classifier learned on the extraction from a set of 1000 Wikipedia sentences labeled as correct or incorrect

• Used features are relation independent

<table>
<thead>
<tr>
<th>Weight</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.16</td>
<td>((x, r, y)) covers all words in (s)</td>
</tr>
<tr>
<td>0.50</td>
<td>The last preposition in (r) is for</td>
</tr>
<tr>
<td>0.49</td>
<td>The last preposition in (r) is on</td>
</tr>
<tr>
<td>0.46</td>
<td>The last preposition in (r) is of</td>
</tr>
<tr>
<td>0.43</td>
<td>(len(s) \leq 10) words</td>
</tr>
<tr>
<td>0.43</td>
<td>There is a WH-word to the left of (r)</td>
</tr>
<tr>
<td>0.42</td>
<td>(r) matches (VW*P) from Figure 1</td>
</tr>
<tr>
<td>0.39</td>
<td>The last preposition in (r) is to</td>
</tr>
<tr>
<td>0.25</td>
<td>The last preposition in (r) is in</td>
</tr>
<tr>
<td>0.23</td>
<td>10 words &lt; (len(s)) (\leq 20) words</td>
</tr>
<tr>
<td>0.21</td>
<td>(s) begins with (x)</td>
</tr>
<tr>
<td>0.16</td>
<td>(y) is a proper noun</td>
</tr>
<tr>
<td>0.01</td>
<td>(x) is a proper noun</td>
</tr>
<tr>
<td>-0.30</td>
<td>There is an NP to the left of (x) in (s)</td>
</tr>
<tr>
<td>-0.43</td>
<td>20 words &lt; (len(s))</td>
</tr>
<tr>
<td>-0.61</td>
<td>(r) matches (V) from Figure 1</td>
</tr>
<tr>
<td>-0.65</td>
<td>There is a preposition to the left of (x) in (s)</td>
</tr>
<tr>
<td>-0.81</td>
<td>There is an NP to the right of (y) in (s)</td>
</tr>
<tr>
<td>-0.93</td>
<td>Coord. conjunction to the left of (r) in (s)</td>
</tr>
</tbody>
</table>

Table 4: ReVERB uses these features to assign a confidence score to an extraction \((x, r, y)\) from a sentence \(s\) using a logistic regression classifier.

Idea: based on labeled examples, weight for each feature is learned. Then a logistic regression classifier is used to combine the weights so to return a value between 0 and 1. Logistic regression function is: \(f(x) = 1/(1+e^{-x})\). In some sense, logistic regression is able to combine weights from any source and can normalize them to interval \([0,1]\). Then, a threshold is used application dependent, e.g., if \(f(x) > 0.5\) then accept else delete new case. In our case, \(x\) would loop across the weights of the activated features.

 Ala Common Lisp:
(defun lr (list &aux (sum (apply #'+ list))) (/ 1.0 (+ 1.0 (exp sum))))
Experiments

• ReVerb is compared to the following systems:

  • ReVerb\textsuperscript{\text{-}lex}: version of ReVerb without lexical constraints

  • TextRunner: extractor of Banko and Etzioni, 2008

  • TextRunner\textsuperscript{R}: TextRunner that uses relation model computed by ReVerb

  • WOE\textsuperscript{pos}: Version of TextRunner using relation learned from Wikipedia by shallow heuristics; developed by Wu and Weld, 2010.

  • WOE\textsuperscript{parse}: Wu and Weld’s parser-based extractor using large set of dependency based extraction patterns.
Test Set

- 500 sentences sampled from Web using Yahoo’s random link service
- Two humans independently evaluated systems' result with 86% agreement
- Uninformative extraction were judged conservatively, e.g., (Ackerman, is a professor of, biology) and (Ackerman, is, a professor of biology) are considered correct.
- For given threshold, precision and recall are computed
  - Precision: fraction of returned extraction that are correct
  - Recall: fraction of correct extractions in the corpus that are returned.
  - To avoid double counting extraction that differ superficially are treated as single extraction (different punctuation, dropping inessential modifiers)
- AUC: precision-recall curves for varying confidence thresholds are considered, and then compute the area under that curve.
Results - AUC and ReVerb-based Systems

Figure 2: REVERB outperforms state-of-the-art open extractors, with an AUC more than twice that of TEXTRUNNER or WOE^{pos}, and 38% higher than WOE^{parse}.

Figure 3: The lexical constraint gives REVERB a boost in precision and recall over REVERB-lex. TEXTRUNNER-R is unable to learn the model used by REVERB, which results in lower precision and recall.
Results - Extractions and Relations Only

ReVerb’s biggest improvement came from the elimination of incoherent extractions.
Result - System Speed

- Run each extractor on a set of 100,000 sentences using a Pentium 4 PC with 4GB ram.

- Processing time:
  - 16 Minutes for ReVerb
  - 21 minutes for TextRunner
  - 21 minutes for WOE$^{\text{pos}}$
  - 11 hours for WOE$^{\text{parse}}$
Error Analysis of Test Corpus

<table>
<thead>
<tr>
<th>REVERB - Incorrect Extractions</th>
<th>REVERB - Missed Extractions</th>
</tr>
</thead>
<tbody>
<tr>
<td>65% Correct relation phrase, incorrect arguments</td>
<td>52% Could not identify correct arguments</td>
</tr>
<tr>
<td>16% N-ary relation</td>
<td>23% Relation filtered out by lexical constraint</td>
</tr>
<tr>
<td>8% Non-contiguous relation phrase</td>
<td>17% Identified a more specific relation</td>
</tr>
<tr>
<td>2% Imperative verb</td>
<td>8% POS/chunking error</td>
</tr>
<tr>
<td>2% Overspecified relation phrase</td>
<td></td>
</tr>
<tr>
<td>7% Other, including POS/chunking errors</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: The majority of the incorrect extractions returned by REVERB are due to errors in argument extraction.

Table 6: The majority of extractions that were missed by REVERB were cases where the correct relation phrase was found, but the arguments were not correctly identified.

- Problems with n-ary relations, e.g., in case ditransitive verbs like „I gave him 15 photographs“; ReVerb extracts only binary relations, e.g., (I, gave, him)

- Improved methods for argument extraction are in order!
Evaluation at Scale

• It is known that frequency of extraction in a large corpus is useful for assessing the correctness of extractions (the more redundant the higher the precision).

• Testing of ReVerb and TextRunner the Corpus of 500 million Web sentences from TextRunner

• It is actually shown that precision increased for more frequent extractions, but that ReVerb obtained precision as TextRunner

• Thus: ReVerb is able to extract more correct extractions at higher precision than TextRunner, even when redundancy is taken into account.
Previous Work for Open IE systems

• Three step approach for binary relation extraction (e.g., TextRunner; Wu and Weld, 2010)

  • Label: sentences are automatically labeled with extractions using heuristics or distant supervision (self-training) - need large set of heuristically labeled examples (e.g., TextRunner up to 200,000 sentences, Wu 300,000

  • Learn: a relation phrase extractor is learned using a sequence-labeling graphical model (e.g., CRF) - training is too expensive

  • Extract: the system takes a sentence as input, identifies pairs of NPs from a sentence, and then use the learned extractor to label each word between the two arguments as part of the relation phrase or not. - used feature functions are not able to cover constraints used by ReVerb or other complex ones
Previous Work for Open IE systems

• Other Web-based approaches avoid relation specific extractions (like on-demand IE or pre-emptive IE) but need document and entity clustering which is too expensive for Web-scale IE.

• Seed-based approaches use existing ontologies, and hence, are too restricted wrt. coverage

• Many systems use full dependency parsing (as we do in DILIA)

• OpenIE is related to semantic role labeling, but this work usually assumes full parsing and hand-crafted semantic resources like FrameNet or ProbBank.
Hybrid Information Extraction

PD Dr. Günter Neumann
DFKI GmbH
Hybrid

- Is a system, if consists of different technologies
- can be combined
- each one depicts a solution by its own
- the integration constitute an innovative plus for the whole system
Examples

hybrid engine

Hybrid Language Processing

HumanMachine
Information Extraction

- The aim of information extraction (IE) is the identification and structuring of domain specific information from free text by skipping irrelevant information at the same time.

- What counts as relevant is given to the system in form of pre-defined domain specific annotations, lexicon entries or rules.
Eine Mixtur aus wachsendem Dienstleistungsgeschäft, Kostensenkungen und erfolgreichen Akquisitionen brachte Wettbewerber IBM im zweiten Quartal deutlich verbesserte Ergebnisse. Zwischen April und Juni stiegen der Umsatz um 10% auf 21,6 Mrd.$ und der Reingewinn auf 1,7 Mrd.$. Sonderlasten in Höhe von 1,4 Mrd.$ hatten den Vorjahresgewinn auf 56 Mill.$ gedrückt.
IE - History

- Early IE-systems were mainly rule-based (manual or learned) and the underlying methodology was specialized for specific applications, cf. MUC systems of the 90th.

- One result of the MUC challenges was a systematic division of labor into IE subtasks
  - Named-Entity Extraction (NER)
  - Relation Entity Extraction (REE)
  - Event Entity Extraction (EEE)
  - Coreferential analysis

The founder of Microsoft, Bill Gates, lives in Seattle, Washington, which is also the place of the company’s headquarter.
IE - the Present

- There exists knowledge-based IE (KIE) and statistical IE (SIE)
- SIE is the State-of-the-Art in research, KIE in industry
- There exists a number of different strategies for the various IE-subtasks
  - from simple gazetteers to complex ontologies
  - from supervised, to minimal supervised to unsupervised Machine Learning algorithms
- Recently, the research focus is on NER, REE, Web-based IE, scalability, domain adaptivity, ...
- Open question: Which method is actually better suited for which text source, domain and application?
Hybrid IE

• Methods and strategies for the combination of different IE-components and the analysis of their plausibility.

• What are possible combinations?
Multi-Strategy

Combiner

IE

IE

IE

IE
Example: NER

Problem:
- Ambiguities
- Bracketing

Solutions:
- meta-learning
- consider IE as independent black-boxes

NER:  
- Wort1
- Wort2
- Wort3
- Wort4
- Wort5

LOC 2
LOC 3
ORG 4
LOC 5
PER 3

Combiner

LingPipe

OpenNLP

BiQue

Sprout
Example: NER

Problem:
- Ambiguities
- Bracketing

Meta learning
- majority voting
- stacking

Strategies:
- maximum weights
- linear regression: \( P_C = 1 - \prod_i (1 - P_i) \)
- cross-validation

Good news:*
Hybrid NER are better than the single NER wrt. recall and precision.

Combining Information Extraction Systems Using Voting and Stacked Generalization
Example - DFKI System NER-Hub

Options

HoverMe for a hint.

paragraph
dummy
opennlp

sentence
dummy
lingpipe
stanford
opennlp_sentence_en

token
dummy
lingpipe
stanford
opennlp_token_en

name
dummy
voting
opennlp_location_en

Processors

Meta Processors

Sending

NER-Hub
- Extraction of Named Entities: Persons, Organizations, Locations, etc.
- Integrates the result of different NE-recognizers (Sprout, OpenNLP, Stanford, LingPipe, etc.)
- Languages: DE, EN
- Implementation: Java, OSGi
Example: Template Filling

Der Gewinn der Schweppes GmbH & Co. KG betrug im ersten Quartal 1997 weit über 20 Mio. DM.

Corpus:
German press releases about turnover (Training: 4850 Tokens, Testing: 1000 Tokens)

Result:
- only MEM: 79.3 %
- only DOP: 51.9 %
- both: 85.2 %

Feature based Strategies

Idea:
- choose a ML algorithm
- choose manually and automatically determined feature templates
- combination of knowledge and statistics

Proposal (Fresko et al., 2005):
- regular grammars (hand coded)
- Maximum Entropy Learning

Co-Training & Bootstrapping

Note: These are manually specified, e.g., through reference to an ontology!

<table>
<thead>
<tr>
<th>Baseline</th>
<th>Co-training</th>
</tr>
</thead>
<tbody>
<tr>
<td>s(the) o(game)</td>
<td>v(win) o(title)</td>
</tr>
<tr>
<td>v(miss) o(game)</td>
<td>s(I) v(play)</td>
</tr>
<tr>
<td>v(play) o(game)</td>
<td>s(miss) o(game)</td>
</tr>
<tr>
<td>v(play) io(in LOC)</td>
<td>s(we) v(play)</td>
</tr>
<tr>
<td>v(go) o(be)</td>
<td>v(lose) o(game)</td>
</tr>
<tr>
<td>s(he) v(be)</td>
<td>s(I) o(play)</td>
</tr>
<tr>
<td>s(that) v(be)</td>
<td>v(make) o(play)</td>
</tr>
<tr>
<td>s(I) v(be)</td>
<td>v(play) io(in game)</td>
</tr>
<tr>
<td>s(it) v(go) o(be)</td>
<td>v(want) o(play)</td>
</tr>
<tr>
<td>s(I) v(think)</td>
<td>v(win) o(MISC)</td>
</tr>
<tr>
<td>s(I) v(know)</td>
<td>s(misc) o(player)</td>
</tr>
<tr>
<td>s(I) v(want)</td>
<td>v(start) o(game)</td>
</tr>
<tr>
<td>s(there) v(be)</td>
<td>s(PER) o(contract)</td>
</tr>
<tr>
<td>s(sc) v(do)</td>
<td>s(we) o(play)</td>
</tr>
<tr>
<td>v(do) o(it)</td>
<td>s(team) v(win)</td>
</tr>
<tr>
<td>s(it) o(be)</td>
<td>v(rush) io(for yard)</td>
</tr>
<tr>
<td>s(sc) v(are)</td>
<td>s(we) o(team)</td>
</tr>
<tr>
<td>s(sc) v(go)</td>
<td>v(win) o(Bowl)</td>
</tr>
<tr>
<td>s(PER) o(DATE)</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Top 20 patterns acquired from the Sports domain by the baseline system (Riloff) and the co-training system for the AP collection. The correct patterns are in bold.

- NER, cf. Singer & Collins, 1999
  Interaction of spelling and context features

- REE, cf. Surdeanu et al. 2006
  Interaction of text classifier and pattern acquisition
QA and Hybrid IE

- Observation: answer extraction is a kind of question-driven IE (NER and REE)

Where does Bill Gates live? lives_in(Town:?, Pers:Bill Gates)

What is a CEO? is_a(Pos:CEO, Conc:?)

Domain open answering of definition questions from the Web

Was ist XYZ? Web-QA

Problem: How to find optimal ranking of answer candidates?

Wikipedia as Blueprint!

- Learn from Wikipedia, what a good verbalization of a definition looks like!

Web-QA

Unsupervised Learning of Feature Model

What is XYZ?

Remark: Method is a step towards Web-scalable ontology learning.
Ontology based IE

The ontology defines the type of the information, which has to be extracted from texts: e.g., types of or institutions and their inter relationship. It defines the structure of the data base, which has to be extracted automatically with the help of OBIE.
TEG - Tree Extraction
Grammars

Manually written extraction grammar CFG

Trained corpus-adapted SCFG

Annotated Corpus

HMM-inspired Semantik Parser

Hand coded grammars

nonterm start Text;
concept Person;
ngram NGFirstName;
ngram NGLastName;
ngram NGNone;
termlist TLHonorific = Mr Mrs Miss Ms <2>Dr;
Person :- <2>TLHonorific NGLastName;
Text :- <11>NGNone Text;
Text :- <2>Person Text;
Text :- <2>;

Parse corpus
adapt rules

Collect statistics

Yesterday, <Person> Dr Simmons </Person>, the distinguished scientist presented the discovery.

$P(Dr \mid TLHonorific) = 1/5$ (choice of one term among five equiprobable ones),
$P(Dr \mid NGFirstName) \approx 1/N$, where $N$ is the number of all known words (untrained ngram behaviour).
TEG - Experiments

MUC-7 NER task

<table>
<thead>
<tr>
<th></th>
<th>HMM entity extractor</th>
<th>Emulation using TEG</th>
<th>DIAL Rules</th>
<th>Full TEG system</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>P</td>
<td>F1</td>
<td>R</td>
</tr>
<tr>
<td>Person</td>
<td>86.91</td>
<td>85.13</td>
<td>86.01</td>
<td>86.31</td>
</tr>
<tr>
<td>Org</td>
<td>87.94</td>
<td>89.75</td>
<td>88.84</td>
<td>85.94</td>
</tr>
<tr>
<td>Location</td>
<td>86.12</td>
<td>87.2</td>
<td>86.66</td>
<td>83.93</td>
</tr>
</tbody>
</table>

ACE-2 relation extraction

<table>
<thead>
<tr>
<th></th>
<th>HMM entity extractor</th>
<th>Markovian SCFG</th>
<th>Full TEG system (with 7 ROLE rules)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recall</td>
<td>Prec</td>
<td>F</td>
</tr>
<tr>
<td>Role</td>
<td>67.55</td>
<td>69.86</td>
<td>68.69</td>
</tr>
<tr>
<td>Person</td>
<td>85.54</td>
<td>83.22</td>
<td>84.37</td>
</tr>
<tr>
<td>Organization</td>
<td>52.62</td>
<td>64.73</td>
<td>58.05</td>
</tr>
<tr>
<td>GPE</td>
<td>85.54</td>
<td>83.22</td>
<td>84.37</td>
</tr>
</tbody>
</table>

INC relation extraction

<table>
<thead>
<tr>
<th></th>
<th>Partial match results</th>
<th>Exact match results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recall</td>
<td>Prec</td>
</tr>
<tr>
<td>PersonAffiliation</td>
<td>89.61</td>
<td>94.52</td>
</tr>
<tr>
<td>OrgLocation</td>
<td>85.32</td>
<td>77.78</td>
</tr>
<tr>
<td>Acquisition</td>
<td>76.00</td>
<td>86.36</td>
</tr>
</tbody>
</table>
TEG - Potential

• Advantages
  • precise rules can be specified for arbitrary IE applications
  • external knowledge sources can be integrated via termlist
  • ngram-context for terminals via ngram (usable for disambiguation)
  • external systems can be integrated
  • "ngram ngOrgNoun featureset ExtPoS restriction Noun;"

• Possible innovations
  • Constraint based formalism as basis for grammar
  • Specialized parsing algorithms (e.g., supertagging)
  • Ontologies as basis for termlist
  • Extending grammars on basis of bootstrapping (human-controlled)
  • ...

Conclusion

- Hybrid IE as innovative plus for IE research and development.
- There exists already a number of promising and exciting approaches.
- High innovation potential to bring language technology, knowledge-based and statistical system under one umbrella.
  - E.g., Multilingual Information Extraction
  - E.g., Multi-Channel Information Extraction