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

Sentence	Incoherent Relation
The guide *contains* dead links and *omits* sites.	contains omits
The Mark 14 *was central* to the *torpedo* scandal of the fleet.	was central torpedo
They *recalled* that Nungesser *began* his career as a precinct leader.	recalled began

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^{parse}'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^{parse}'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

$$V \mid V P \mid V W^{*} P$$

V = verb particle? adv?
W = (noun | adj | adv | pron | det)
P = (prep | particle | inf. marker)

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 !

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

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 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 show k=20 as 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

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

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^{-lex}: version of ReVerb without lexical constraints
- TextRunner: extractor of Banko and Etzioni, 2008
- TextRunner^{-R}: TextRunner that uses relation model computed by ReVerb
- WOE^{pos}: Version of TextRunner using relation learned from Wikipedia by shallow heuristics; developed by Wu and Weld, 2010.
- WOE^{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 are under that curve.

Results - AUC and ReVerb-based Systems

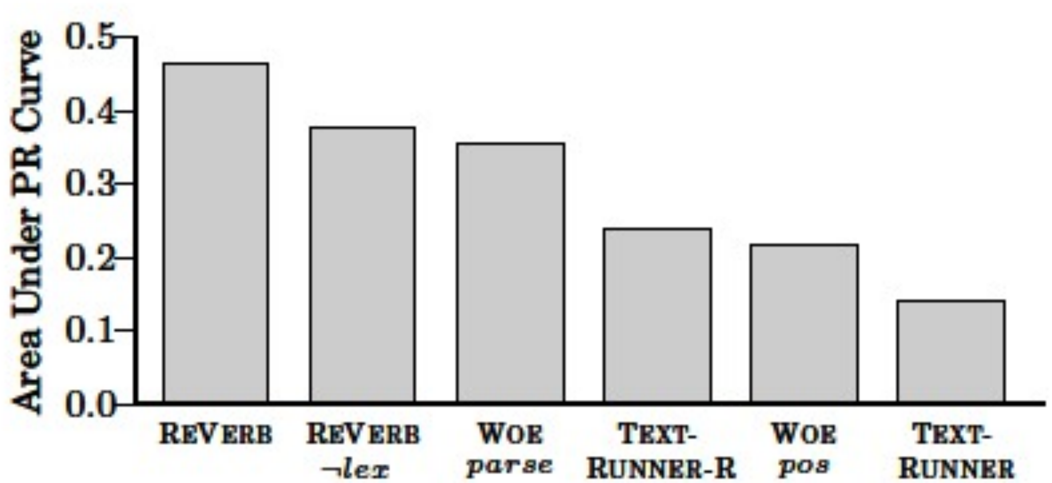


Figure 2: REVERB outperforms state-of-the-art open extractors, with an AUC more than twice that of TEXTRUNNER or WOE^{pos}, and 38% higher than WOE^{parse}.

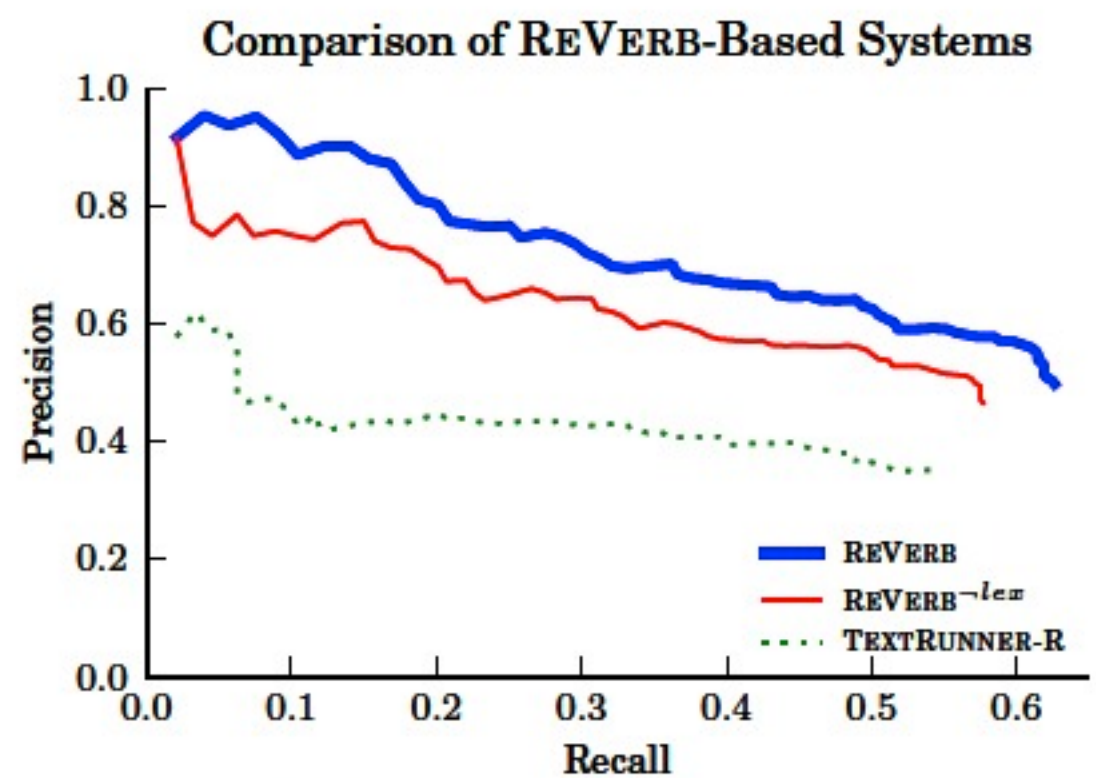


Figure 3: The lexical constraint gives REVERB a boost in precision and recall over REVERB$^{\neg lex}$. TEXTRUNNER-R is unable to learn the model used by REVERB, which results in lower precision and recall.

Results - Extractions and Relations Only

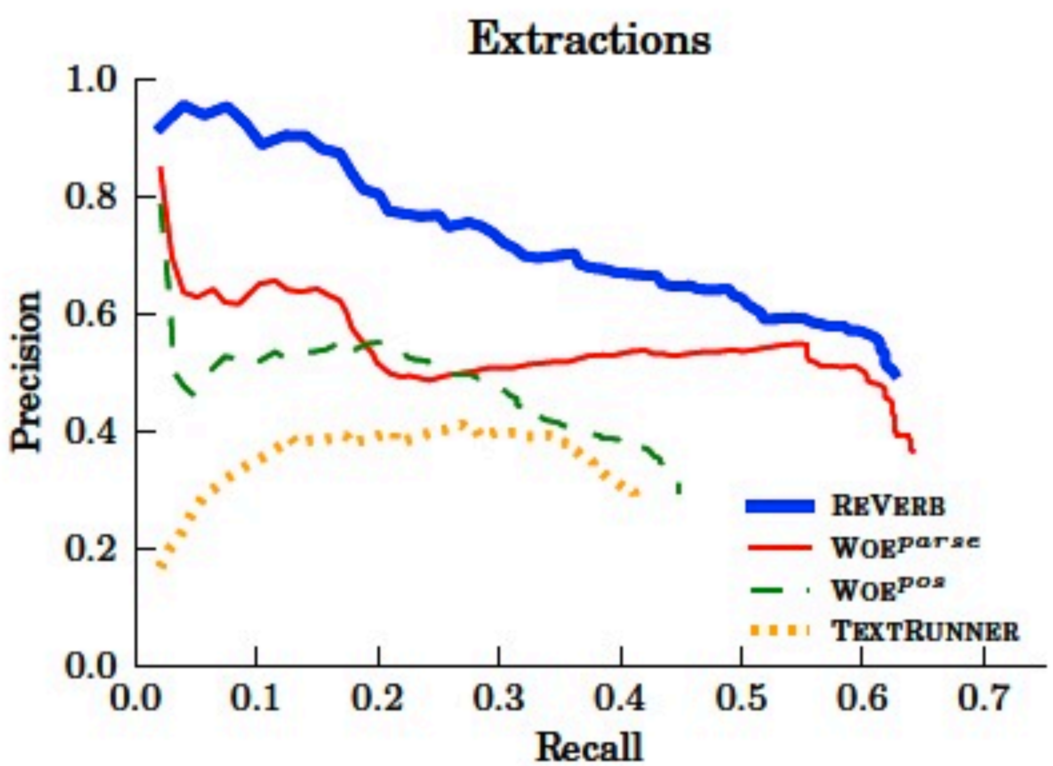


Figure 4: REVERB achieves significantly higher precision than state-of-the-art Open IE systems, and comparable recall to WOE^{parse}.

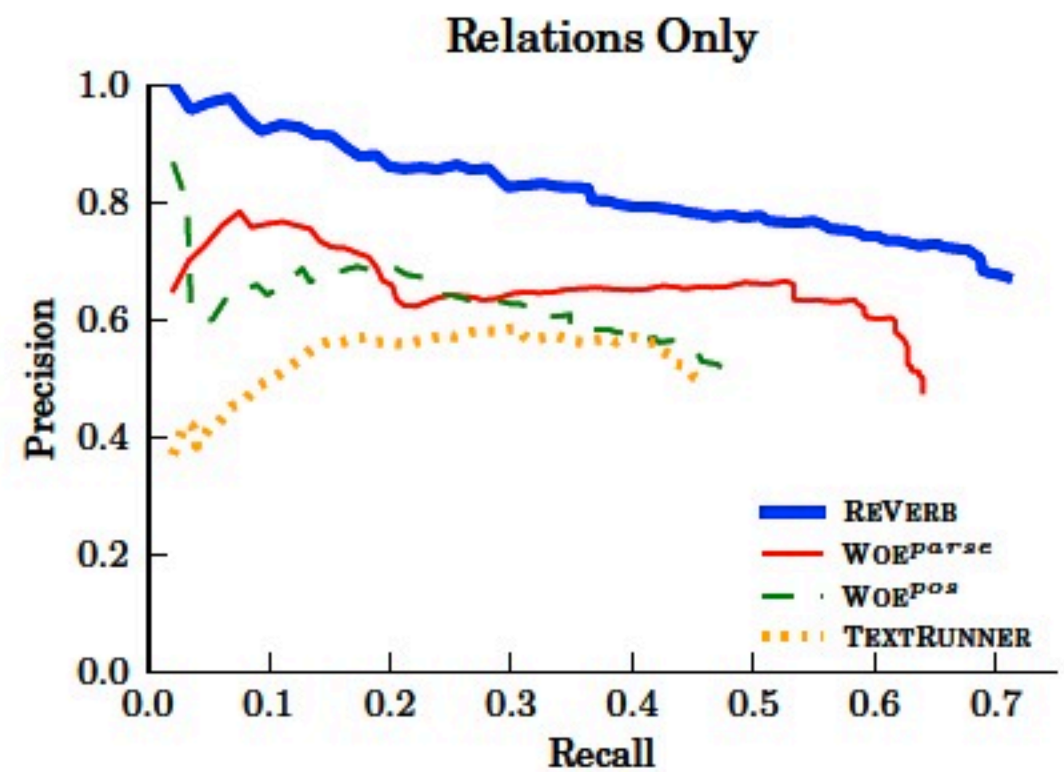


Figure 5: On the subtask of identifying relations phrases, REVERB is able to achieve even higher precision and recall than other systems.

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^{pos}
  - 11 hours for WOE^{parse}

Error Analysis of Test Corpus

ReVerb - Incorrect Extractions	
65%	Correct relation phrase, incorrect arguments
16%	N-ary relation
8%	Non-contiguous relation phrase
2%	Imperative verb
2%	Overspecified relation phrase
7%	Other, including POS/chunking errors

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

ReVerb - Missed Extractions	
52%	Could not identify correct arguments
23%	Relation filtered out by lexical constraint
17%	Identified a more specific relation
8%	POS/chunking error

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 to 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

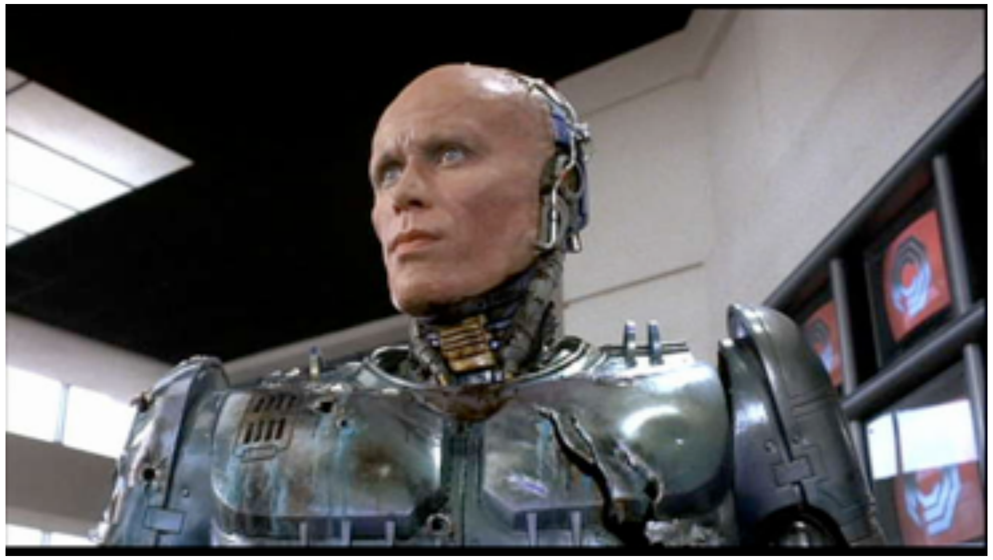
Hybrid

- 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

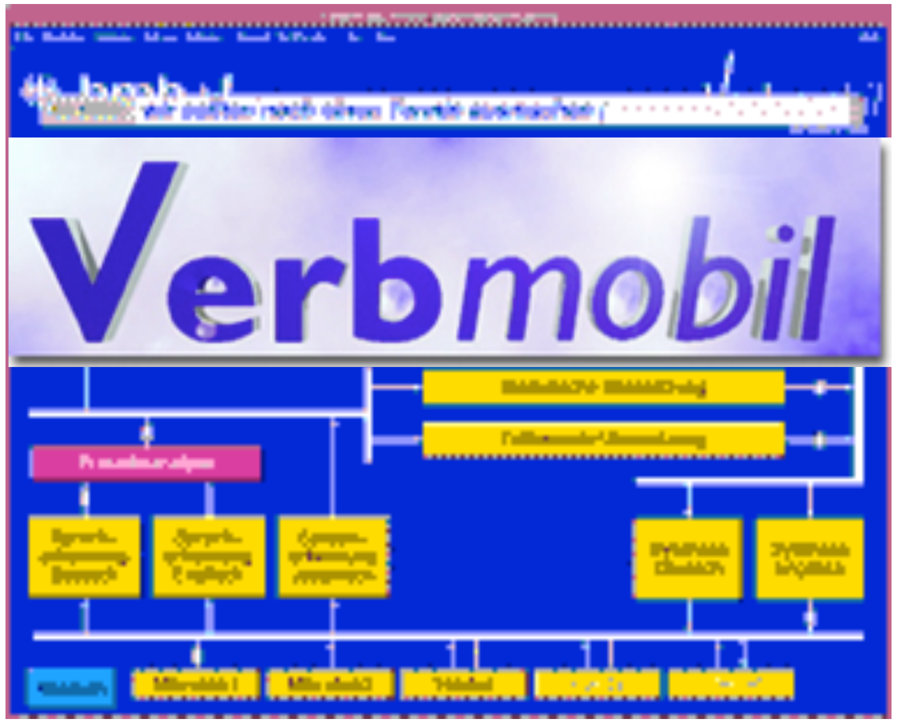
Examples

hybrid engine

HumanMachine



Hybrid Language Processing



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.

Example: news about turnover

turnover(Company, Year, Manner, Amount, Tendendcy, Differnce)

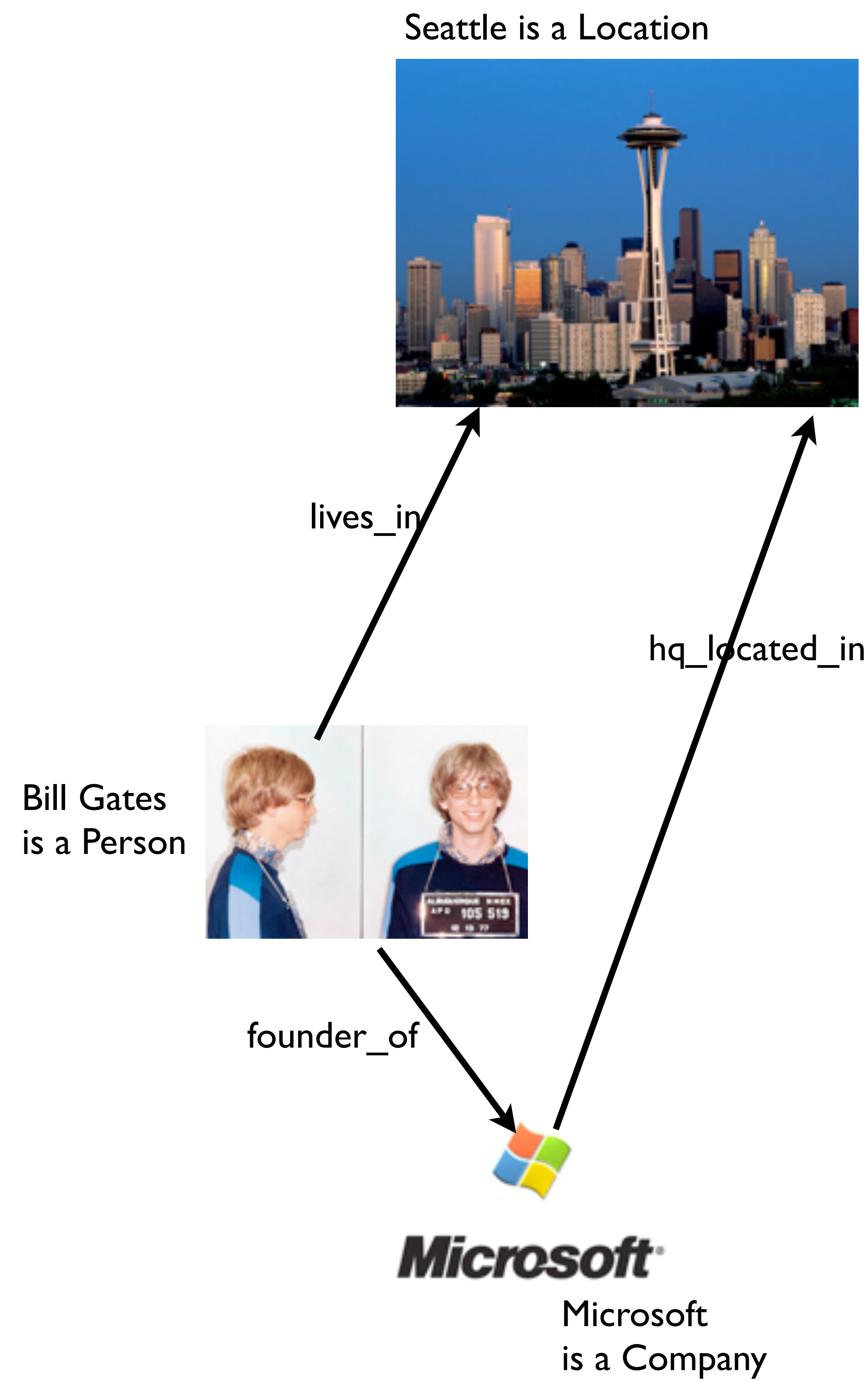
Unternehmen	Jahr	Größe	Betrag	Tendenz	Differenz
Compaq	1998	Umsatz	31 Mrd. USD	+	27%

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.

Unternehmen	Jahr	Größe	Betrag	Tendenz	Differenz
IBM	2003	Umsatz	21,6 Mrd. $	+	10 %

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.

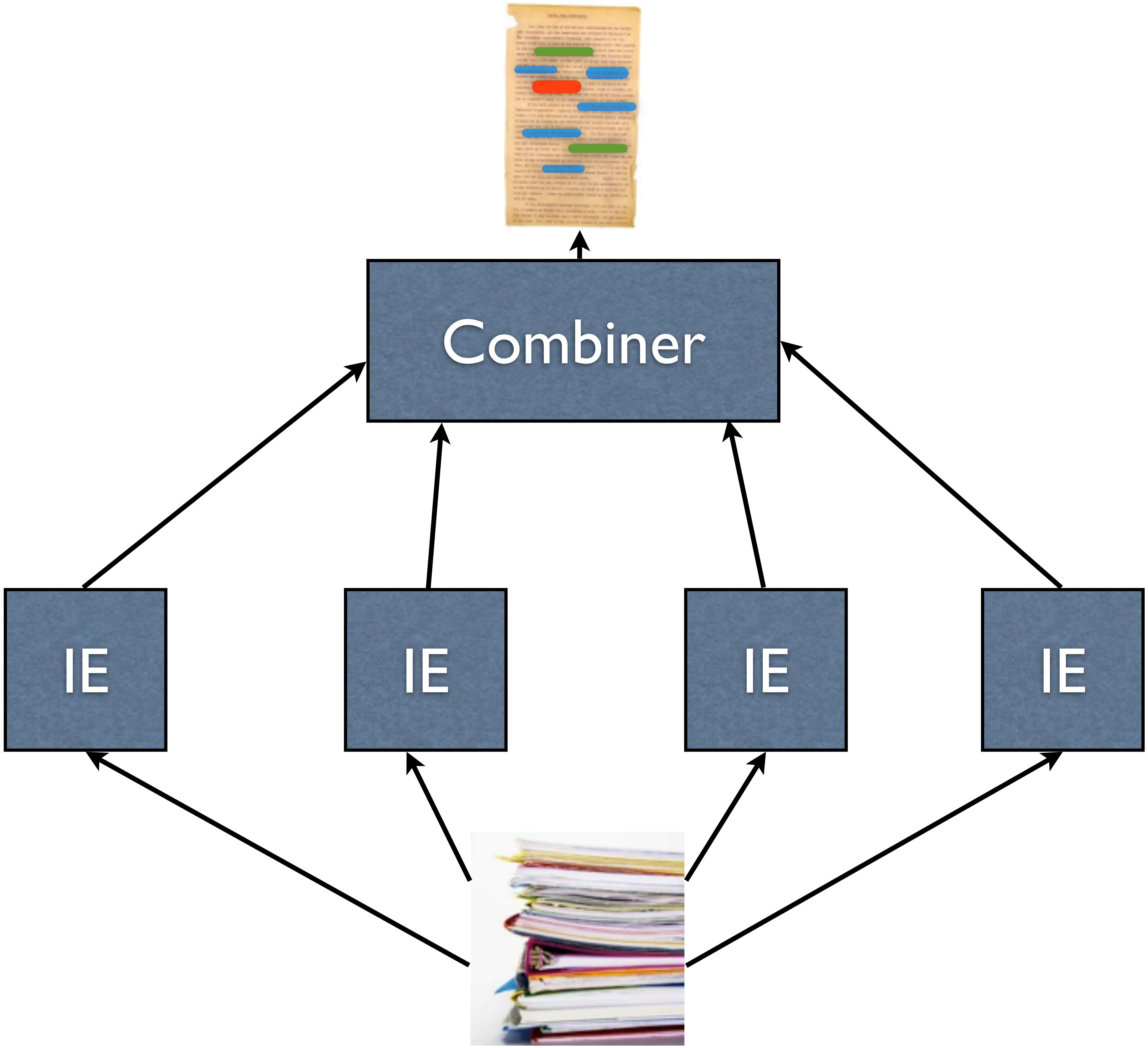
IE - the Present

- 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?

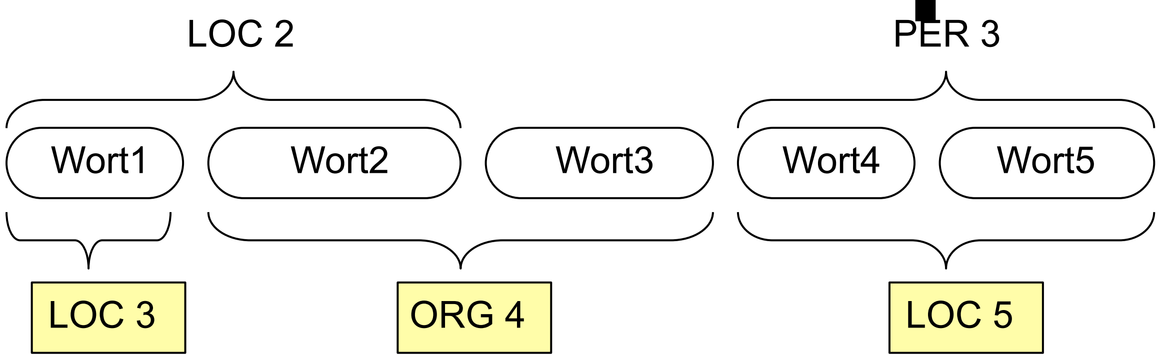
Hybrid IE

- Methods and strategies for the combination of different IE-components and the analysis of their plausibility.
- What are possible combinations ?

Multi-Strategy

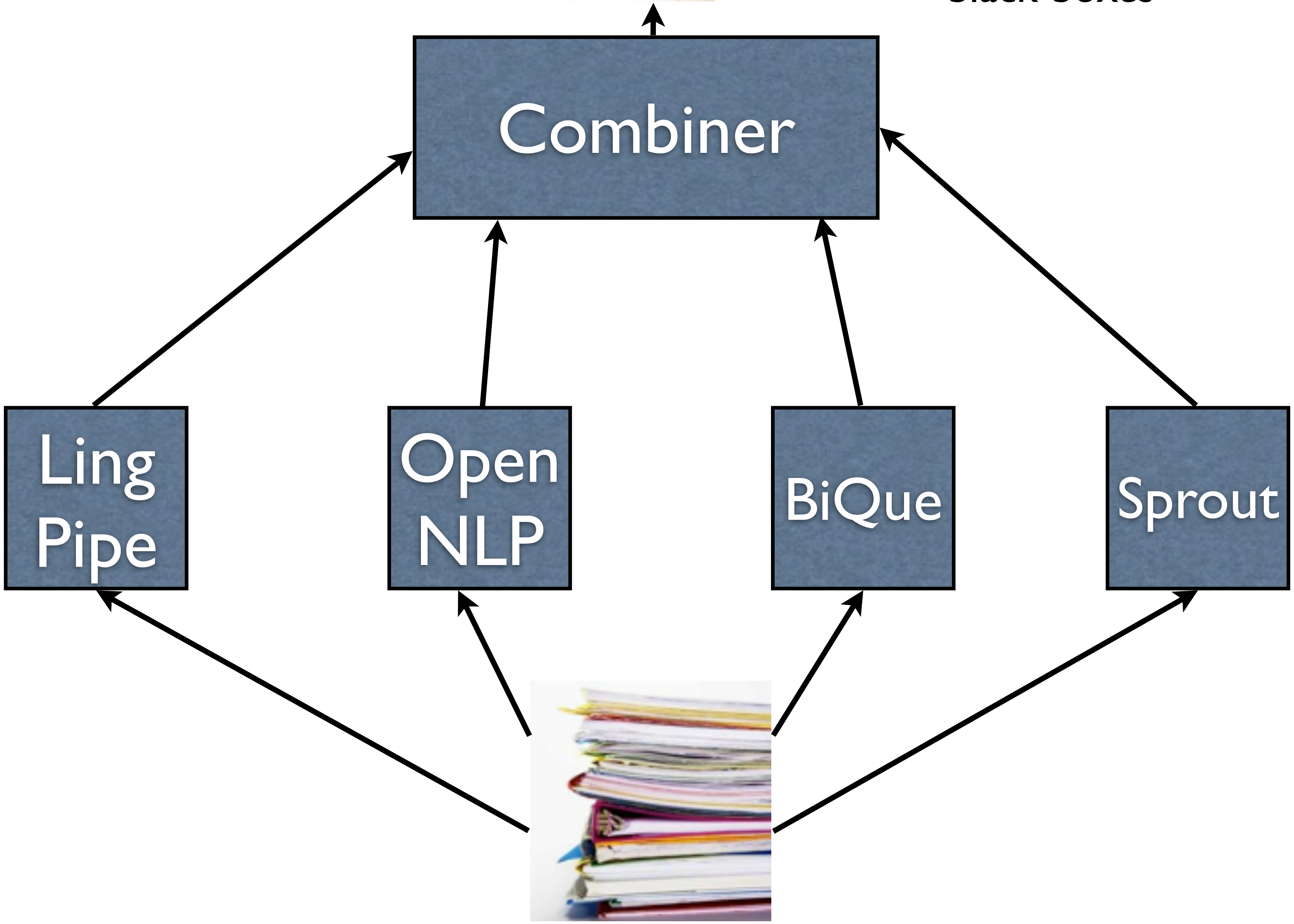


Example: NER

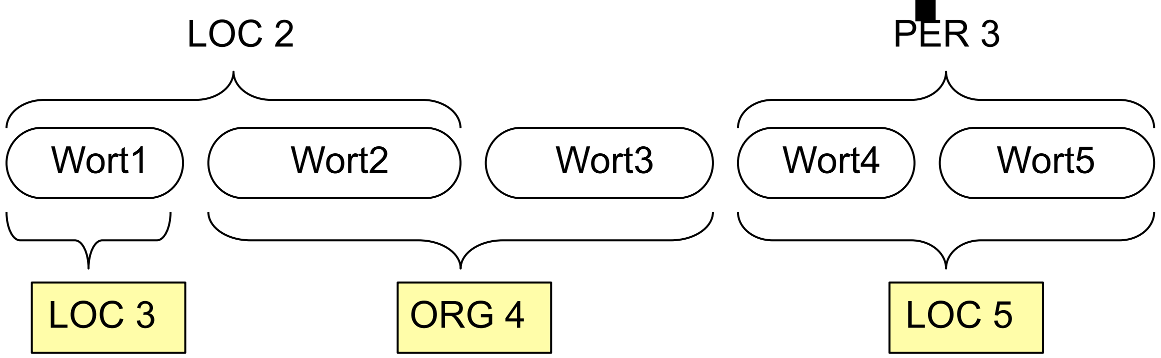


Solutions:
- meta-learning
- consider IE_i as independent black-boxes

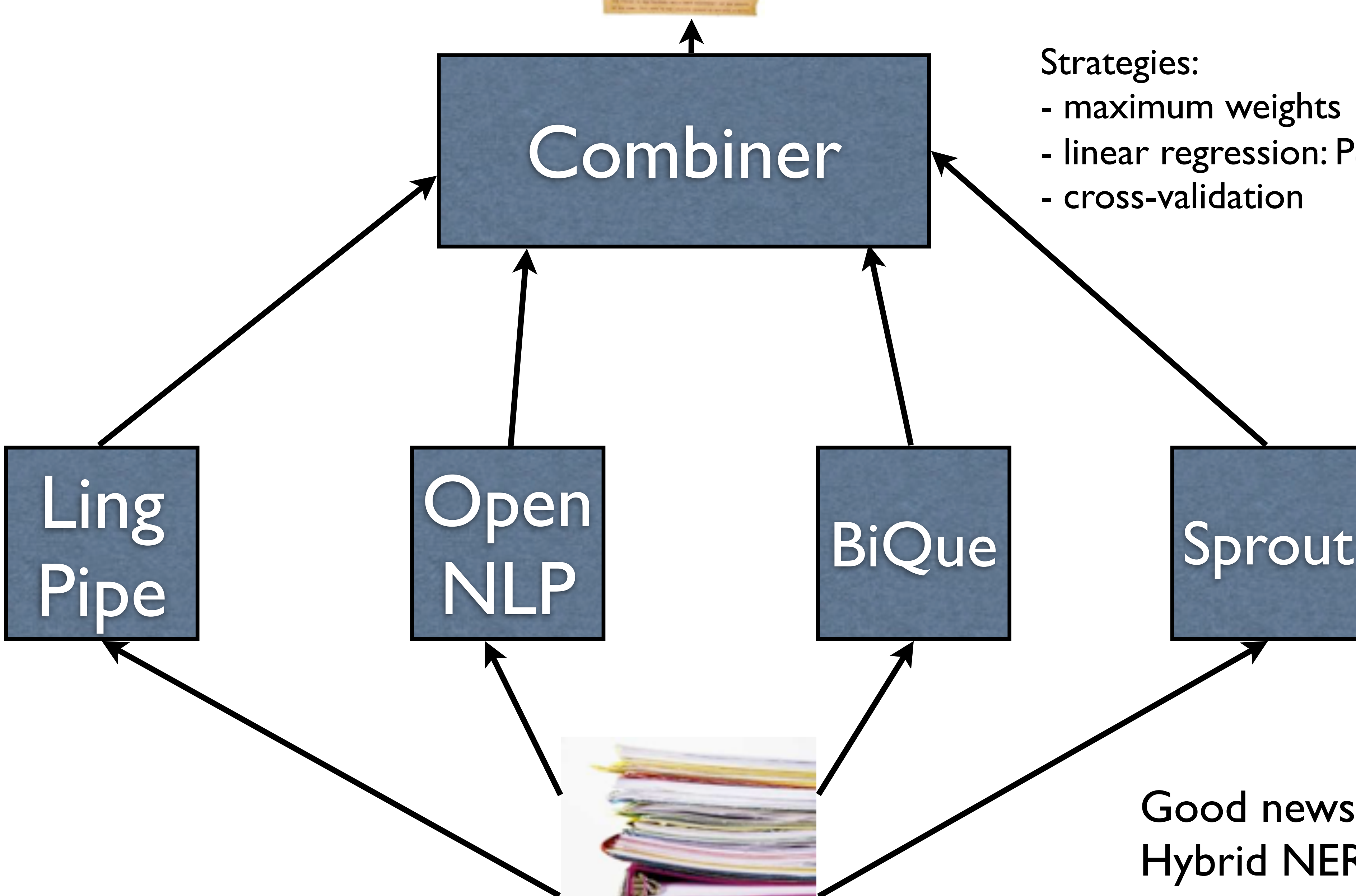
Problem:
- Ambiguities
- Bracketing



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
by: G Sigletos et al., J. Mach. Learn. Res., Vol. 6 (2005), pp. 1751-1782.

Example - DFKI System NER-Hub

Selection and orchestration of the components

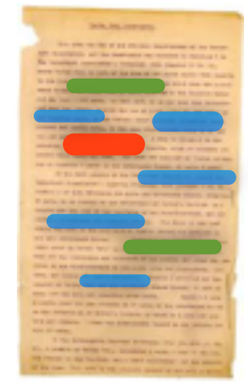
Voting mechanism

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 <Org>der Schweppes Gmbh & Co.</Org> KG
betrug <TIMEX>im ersten Quartal 1997</TIMEX> weit ueber 20 Mio. DM.



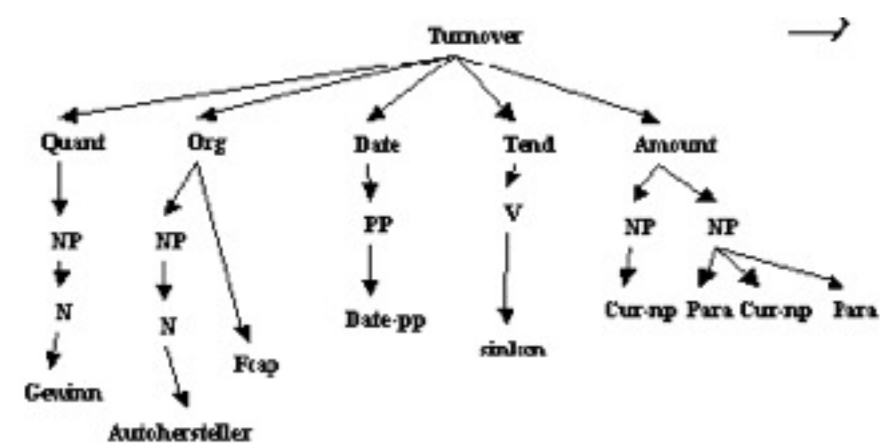
Result:
- only MEM: 79.3 %
- only DOP: 51.9 %
- both: 85.2 %

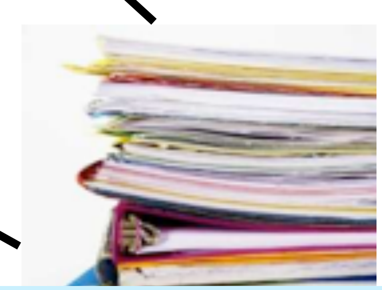
Corpus:
German press releases about turnover (Training: 4850 Tokens, Testing: 1000 Tokens)

Iterative Tag Insertion

MEM - Maximum Entropy Modeling

DOP - Data-Oriented Parsing

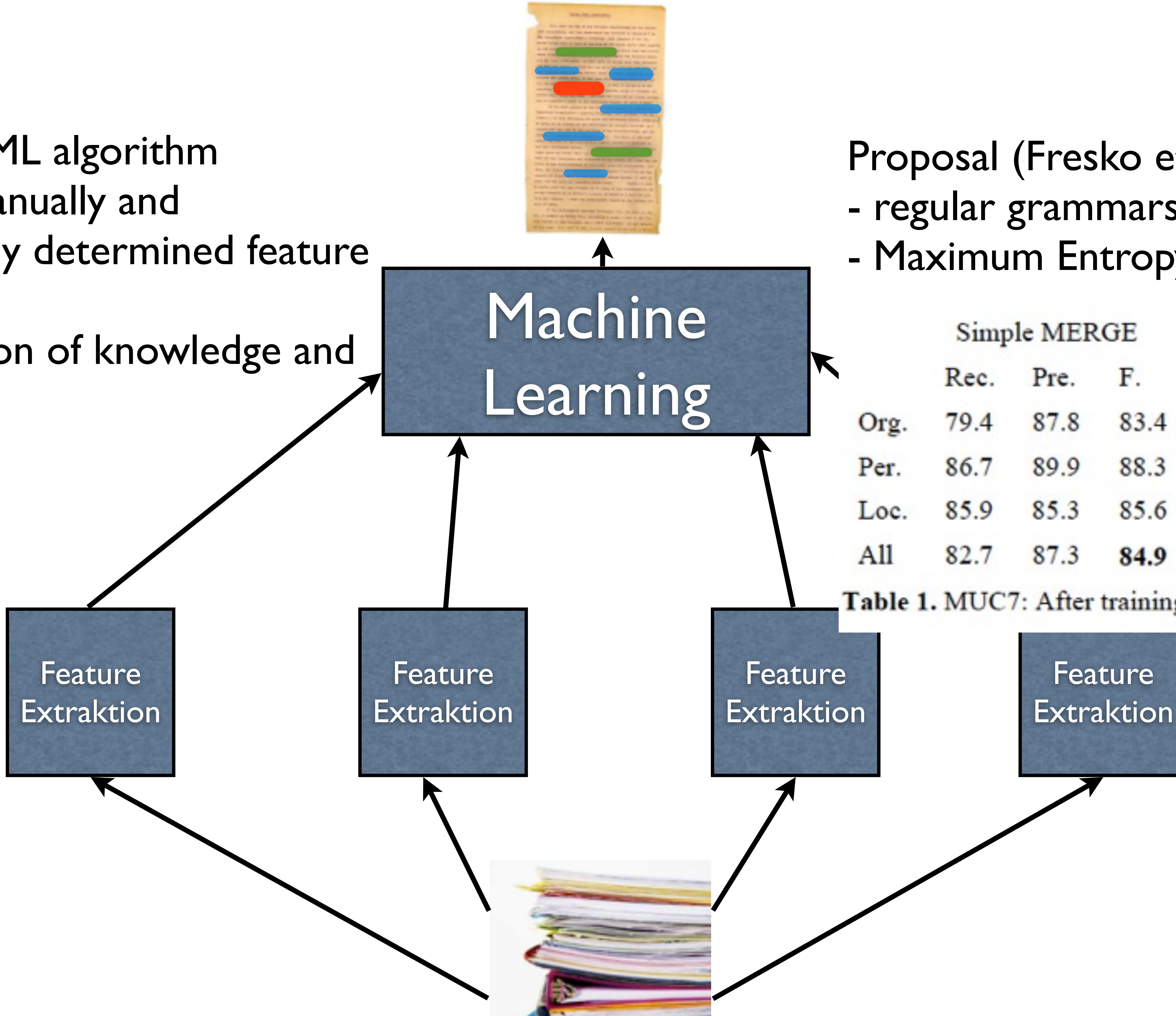




Neumann, G. (2006) A Hybrid Machine Learning Approach for Information Extraction from Free Texts. In Spiliopoulou at al. (Eds). From Data and Information Analysis to Knowledge Engineering, Springer series: Studies in Classification, Data Analysis, and Knowledge Organization, pages 390-397, Springer-Verlag Berlin, Heidelber, New-York.

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

	Simple MERGE			MERGE+Rules		
	Rec.	Pre.	F.	Rec.	Pre.	F.
Org.	79.4	87.8	83.4	83.9	90.9	87.2
Per.	86.7	89.9	88.3	90.6	93.4	92.0
Loc.	85.9	85.3	85.6	91.5	91.8	91.7
All	82.7	87.3	**84.9**	87.9	91.7	**89.8**

Table 1. MUC7: After training with 100 documents.

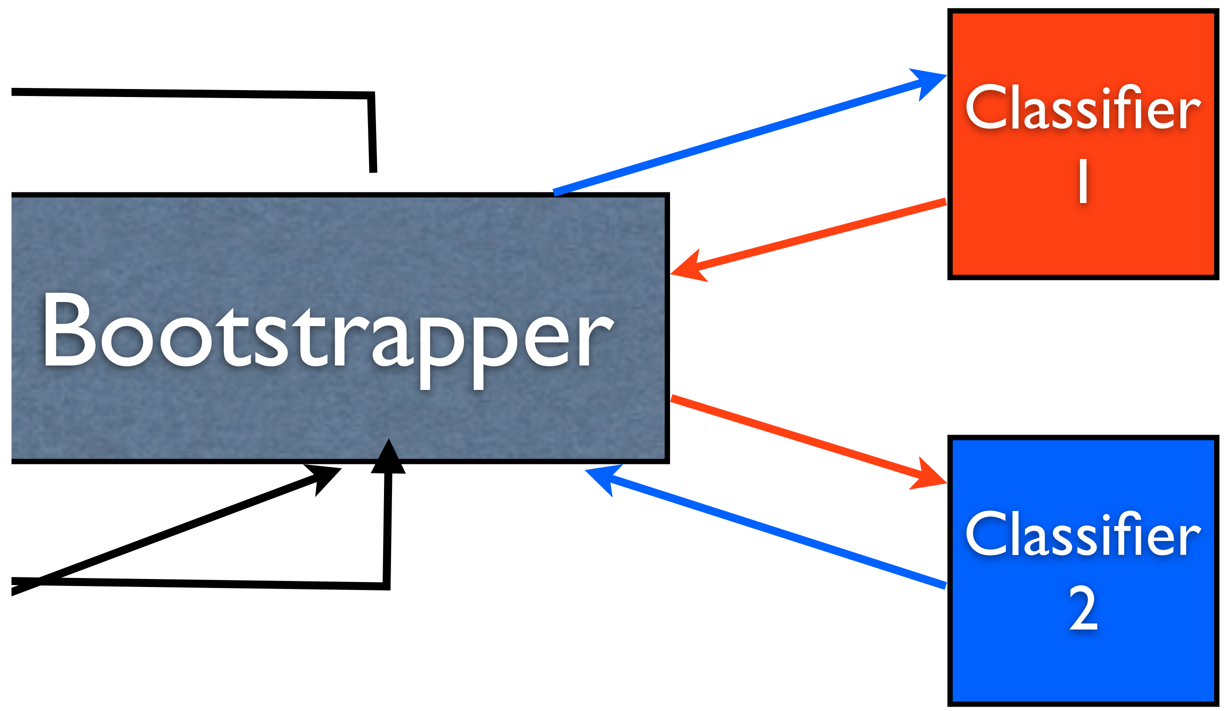
Fresko, Rozenfeld, Feldman „A Hybrid Approach to NER by Integrating Manual Rules into MEM“, CIKM, 2005.

Co-Training & Bootstrapping

Note:
These are
manually
specified, e.g.,

Baseline	Co-training
s(he) o(game)	**v(win) o(title)**
v(miss) o(game)	s(I) v(play)
v(play) o(game)	**s(he) v(game)**
v(play) io(in LOC)	s(we) v(play)
v(go) o(be)	**v(miss) o(game)**
s(he) v(be)	**s(he) v(coach)**
s(that) v(be)	**v(lose) o(game)**
s(I) v(be)	s(I) o(play)
s(it) v(go) o(be)	v(make) o(play)
s(it) v(be)	**v(play) io(in game)**
s(I) v(think)	v(want) o(play)
s(I) v(know)	**v(win) o(MISC)**
s(I) v(want)	**s(he) o(player)**
s(there) v(be)	**v(start) o(game)**
s(we) v(do)	s(PER) o(contract)
v(do) o(it)	s(we) o(play)
s(it) o(be)	**s(team) v(win)**
s(we) v(are)	**v(rush) io(for yard)**
s(we) v(go)	**s(we) o(team)**
s(PER) o(DATE)	**v(win) o(Bowl)**

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.



ing & IE

f 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:?)

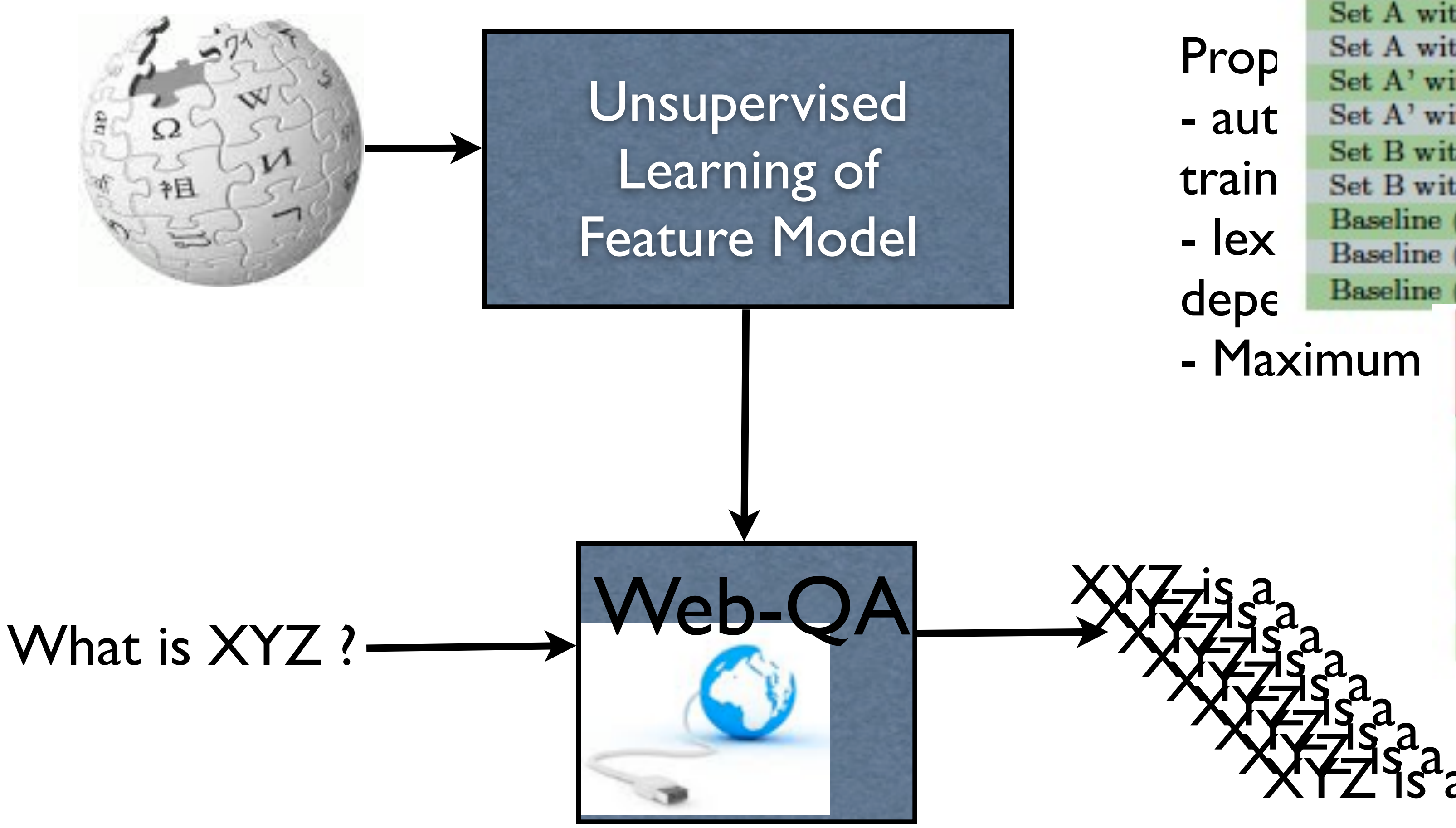
Domain open answering of definition questions from the Web

Problem:
How to find optimal ranking of answer candidates?

Figueroa, A., Neumann, G. and Atkinson, J. (2009) Searching for Definitional Answers on the Web using Surface Patterns. In journal IEEE Computer volume 42 number 4, Pages 68-76, IEEE, 4/2009.

Wikipedia as Blueprint!

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



Best Attributes Found for	Applied to Set A' Accuracy	Applied to Set A Accuracy
Set A without NLP	81.16%	
Set A with NLP	**85.94%**	
Set A' without NLP		78.28%
Set A' with NLP		**83.04%**
Set B without NLP	76.86%	75.50%
Set B with NLP	63.19%	74.61%
Baseline (0.3)	54.13%	44.17%
Baseline (0.2)	54.63%	44.23%
Baseline (0.1)	56.25%	44.77%

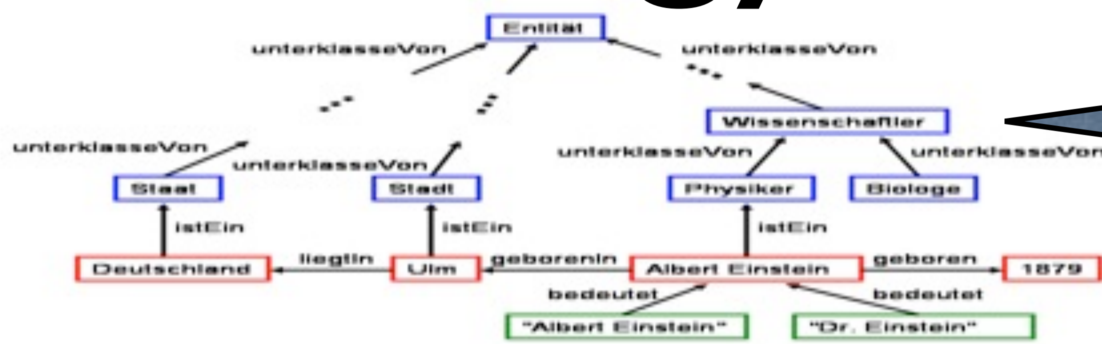
Best Attributes Found for	Applied to Set B Accuracy
Set A without NLP	**59.86%**
Set A with NLP	58.71%
Set A' without NLP	58.25%
Set A' with NLP	58.44%
Baseline (0.3)	56.26%
Baseline (0.2)	57.77%
Baseline (0.1)	56.73%

Prop
- aut
train
- lex
depe
- Maximum

according to similarity of
Wikipedia?

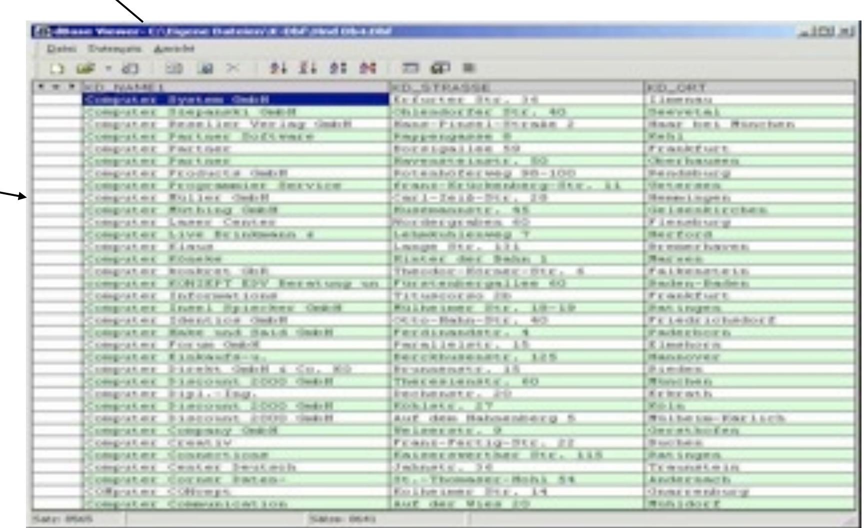
Remark: Method is a step towards Web-scalable ontology learning.

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.

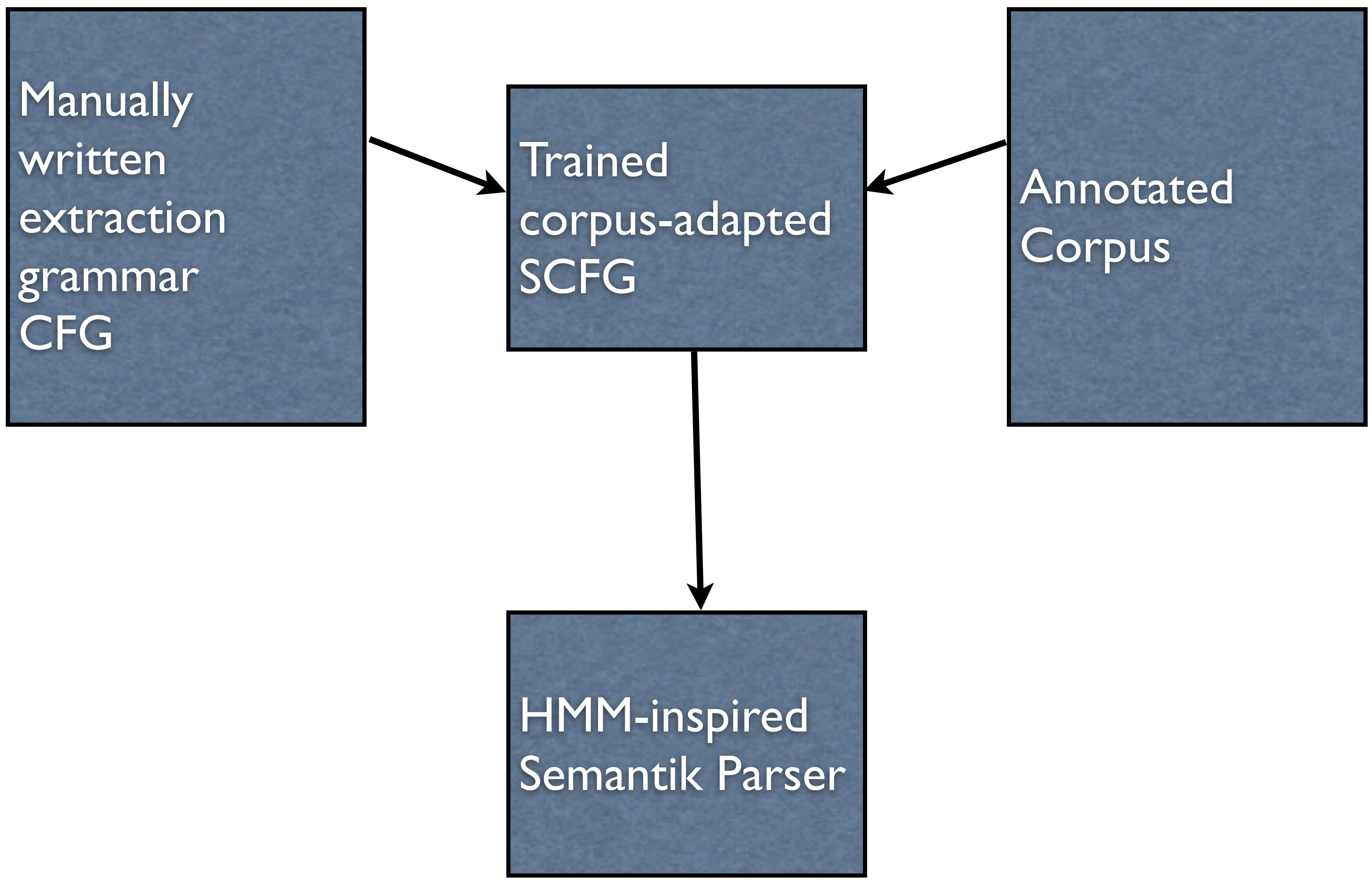
IE



Ontology population
ontology learning

THESEUS
Projekt TheseusOrdo
TechWatch

TEG - Tree Extraction Grammars



Rosenfeld, Feldman & Freski „TEG - a hybrid approach to information extraction“, Knowledge Information Systems (2006) 1-18.

TEG - Example

Hand coded grammars

nonterm start Text;
concept Person;
ngram NGFirstName;
ngram NGLastName;
ngram NGNone;
termlist TLHonorific = Mr Mrs Miss Ms Dr;
(1) Person :- TLHonorific NGLastName;
(2) Person :- NGFirstName NGLastName;
(3) Text :- NGNone Text;
(4) Text :- Person Text;
(5) Text :- ;

termlist TLHonorific = Mr Mrs Miss Ms <2>Dr;
Person :- <2>TLHonorific NGLastName;
Text :- <11>NGNone Text;
Text :- <2>Person Text;
Text :- <2>;

adapt rules

P(Dr | **TLHonorific***)* = 1/5 (choice of one term among five equiprobable ones),
P(Dr | **NGFirstName***)* ≈ 1/*N*, where *N* is the number of all known words (untrained ngram behaviour).

Parse corpus

Collect statistics

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

TEG - Experiments

MUC-7 NER task

	HMM entity extractor			Emulation using TEG			DIAL Rules			Full TEG system		
	R	P	F1	R	P	F1	R	P	F1	R	P	F1
Person	86.91	85.13	86.01	86.31	86.83	86.57	81.32	93.75	87.53	93.75	90.78	92.24
Org	87.94	89.75	88.84	85.94	89.53	87.7	82.74	93.36	88.05	89.49	90.9	90.19
Location	86.12	87.2	86.66	83.93	90.12	86.91	91.46	89.53	90.49	87.05	94.42	90.58

ACE-2 relation extraction

	HMM entity extractor			Markovian SCFG			Full TEG system (with 7 ROLE rules)		
	Recall	Prec	F	Recall	Prec	F	Recall	Prec	F
Role				67.55	69.86	68.69	83.44	77.3	80.25
Person	85.54	83.22	84.37	89.19	80.19	84.45	89.82	81.68	85.56
Organization	52.62	64.735	58.05	53.57	67.46	59.71	59.49	71.06	64.76
GPE	85.54	83.22	84.37	86.74	84.96	85.84	88.83	84.94	86.84

INC relation extraction

	Partial match results			Exact match results		
	Recall	Prec	F	Recall	Prec	F
PersonAffiliation	89.61	94.52	**92.00**	75.33	79.46	**77.33**
OrgLocation	85.32	77.78	**80.00**	76.47	72.22	**74.29**
Acquisition	76.00	86.36	**80.85**	68.00	77.27	**72.34**

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