Preemptive & On-Demand IE

Andrea Heyl

Saarland University

January, 18, 2007

Preemptive & On-Demand Information Extraction

Motivation

- Present situation: adapting IE systems to new topics requires lots of time and human effort
  - find relevant frames for the topic
  - create patterns to extract these frames
  - often uses annotated corpora
- Vision of On-Demand/Preemptive IE:
  "Create all feasible IE systems in advance"
  - let the user determine the topic through their query
  - find relevant sentence patterns
  - use these patterns to create salient relations
  - high portability

Preemptive & On-Demand Information Extraction

Motivation

Article 1:

"Yesterday, hurricane Katrina hit the coast of New Orleans."

Article 2:

"Longwang headed towards Taiwan."

	hurricane	target
article 1	Katrina	New Orleans
article 2	Longwang	Taiwan

Preemptive & On-Demand Information Extraction

Motivation

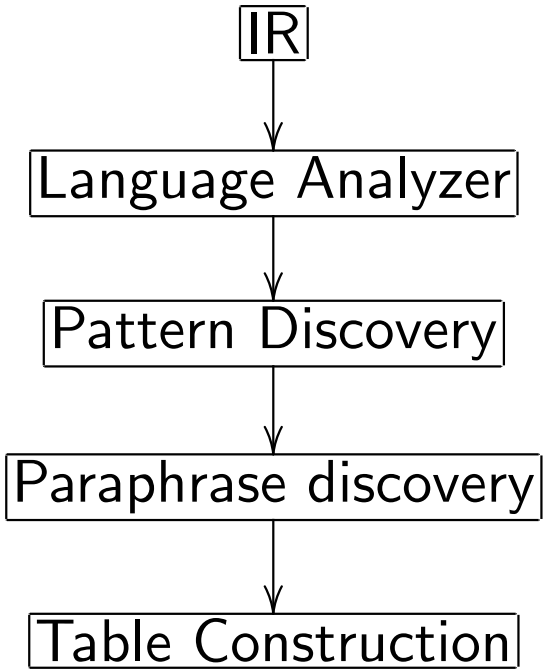
- ODIE/PIE *can*
  - automatically adapt to new topics
  - identify the most salient structures
  - extract information on the users demands
- ODIE/PIE *uses*
  - unsupervised learning methods (no labeled training data)
  - basic NLP tools (POS tagging, NE tagging...)

Preemptive & On-Demand Information Extraction

Basic Idea: Creating Tables

- find the NEs in texts and the "basic patterns" connected to them
- connect entities that share a basic pattern
- find multiple parallel correspondences between articles
  - Article A: ...Katrina headed ... New Orleans was hit...
  - Article B: ...Longwang headed ... Taiwan was hit...
- put articles that have similar relations in the same cluster

Dataflow


IR
Language Analyzer
Pattern Discovery
Paraphrase discovery
Table Construction

IR part

- User describes the topic of interest in keyword(s), e.g. “hurricanes”
- Relevant documents are retrieved using the query

Language Analyzing Tools

- extended NE set with more that 140 hierarchical types
- use only those NE instances as slot fillers in the tables
- no coreference resolution yet

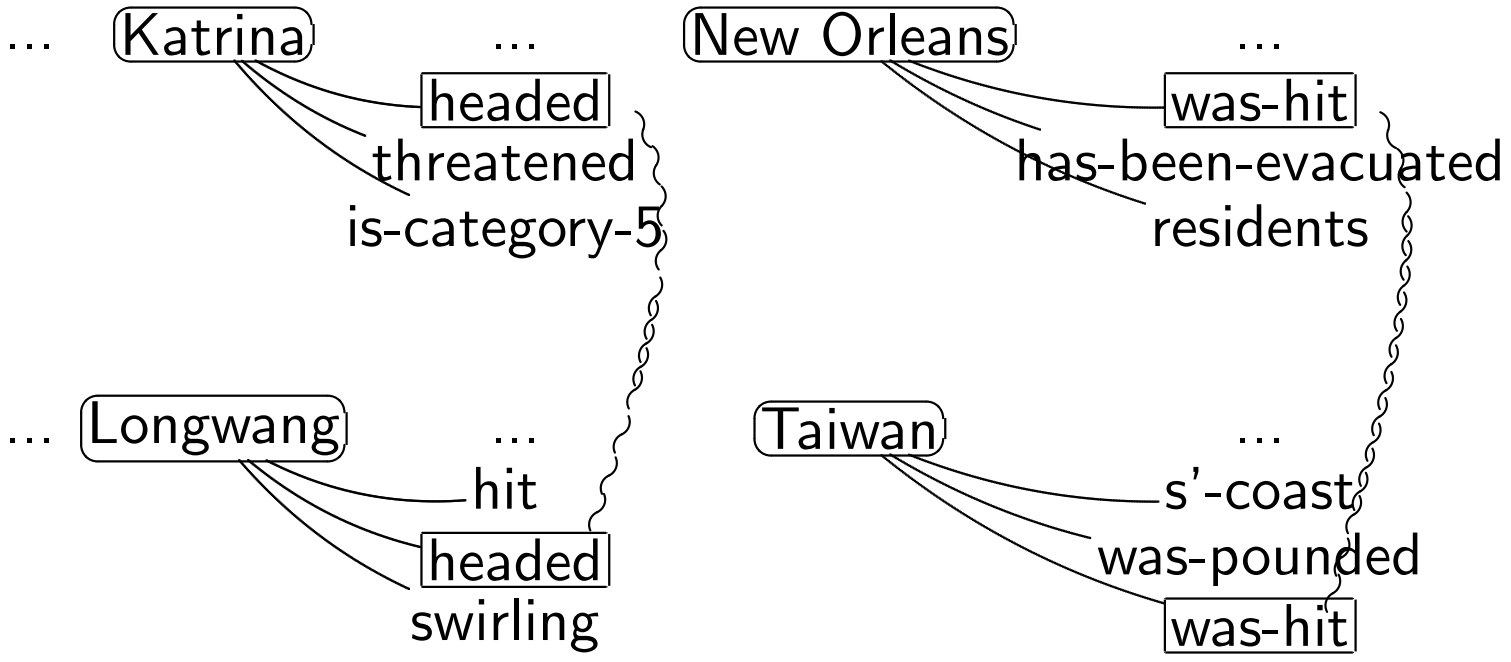
Basic Patterns

- in general: a part of the text that is syntactically connected to a NE, e.g. New Orleans *is hit*
- "role" of an entity, e.g. the place affected by a hurricane
- Problem: What counts as a pattern?
  How to get useful patterns? (*the* hurricane?)

Basic Patterns

- build patterns on a predicate-argument structure
- dependency trees
- a pattern is: every subtree of the dependency tree for a sentence that contains a verb
- Pattern ranking:
  $score(t : subtree) = \frac{\text{frequency of t in retrieved docs}}{log(\text{global frequency of t})}$

Basic Patterns



article A	Katrina	New Orleans
article B	Longwang	Taiwan

Problems?

- *Katrina flew to New Orleans* vs. *President Bush flew to New Orleans*
  $\Longrightarrow$ use bag-of-words approach
- exclude frequent pattern like *say* or *have*
- increase the number of basic patterns
  $\Longrightarrow$ basic clustering

Paraphrase Discovery

- "link patterns that mean the same thing"
- idea: if two NEs occur pairwise in different sentences, those sentences are likely to have the same meaning
- $<$ COM$_1$ $>$<agree to buy>$<$ COM$_2$ $>$<for MNY>
  $<$ COM$_1$ $>$<will acquire>$<$ COM$_2$ $>$<for MNY>
  <a MNY merger><of COM$_1$ ><and COM$_2$ >
- cluster the phrases from such sentences using keywords
- put paraphrases into the same pattern set

Table Construction

- apply the pattern set to the original corpus
- create one table for each pattern set
- delete tables with less than three rows
- **Pattern Set**

  * COM_1 agree to buy COM_2 for MNY
  * a MNY merger of COM_1 and COM_2

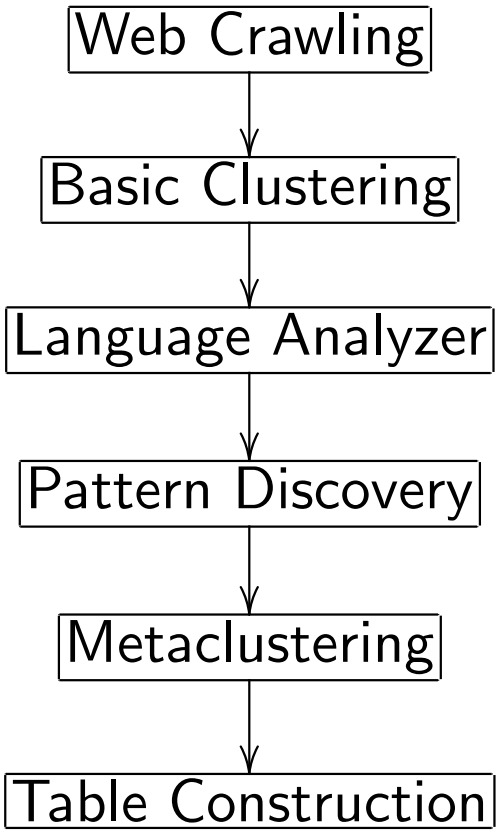
  Newspaper

  $article_1$: ABC agreed to buy CDE for \$1M
  $article_2$: a \$20M merger of FGH and IJK

  Constructed table

Article	Company	Money
1	ABC, CDE	\$1M
2	FGH, IJK	\$20M

Dataflow

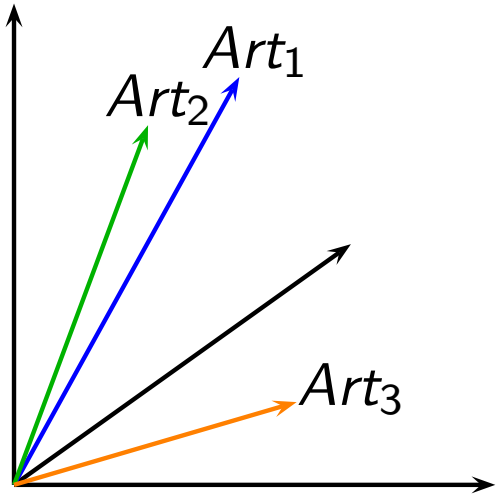

Web Crawling
Basic Clustering
Language Analyzer
Pattern Discovery
Metaclustering
Table Construction

Web Crawling

- no topics are given
- extract articles from different news sites

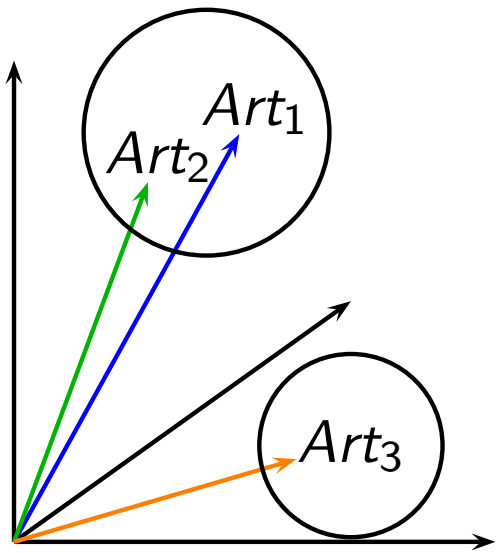
Basic Clustering

- Task: find articles that talk about the same event
- compute a word vector for each article
- compute similarity between all possible pairs of articles
- cluster those articles whose similarity exceeds a certain threshold



Basic Clustering

- Task: find articles that talk about the same event
- compute a word vector for each article
- compute similarity between all possible pairs of articles
- cluster those articles whose similarity exceeds a certain threshold



Language Analyzing Tools

- use only "classical" NEs: *Person*, *Organization*, *GPE*, *Location and Facility*
- coreference resolution within each article
- connect entities in different articles (string matching)
- use GLARF structures instead of dependency trees

Pattern discovery

- find patterns for the five most frequent entities in each cluster
- take the smallest patterns that contain a predicative word
- replace NEs ocurrences by their NE type, e.g. “PER hit GPE”

Metaclustering

- give weights to the patterns:
  $weight(p) = -log \frac{\text{clusters that include p}}{\text{all clusters}}$
- for all pairs of basic clusters A1 and A2:
  - compute pattern similarity between A1 and A2 as the weighted sum of all shared patterns
  - compute the word similarity between A1 and A2
  - merge A1 and A2 if both similarities exceed a threshold
- build a table for every metacluster

Table Construction

- **Articles in a MetaCluster**
  - ...as **Rita** slammed headed west, perhaps towards **Texas**...
  - ...Typhon **Damrey** slammed into **Vietnam** on Tuesday killing...
  - Oil markets have been watching **Wilma**'s progress...the **storm** will turn towards **Florida**
  - President **Bush** flew to **Texas** last Wednesday...

Constructed table

Article	hurricane	coast
2005-09-21	Rita	Texas
2005-09-27	Damrey	Vietnam
2005-10-19	Wilma	Florida
2005-09-25	Bush	Texas

Evaluation

- no “gold standard”
- most IE tasks are evaluated by human judgement
- “intuitive” evaluation criteria

Evaluation
(PIE)

- Human evaluator tries to find a name for the relation(s) expressed in a table
- Table is consistent if at least half of the rows fit the explanation.

Consistent tables	36 (75%)
Inconsistent tables	12
Total	48

Rows that fit the description	118 (73%)
Rows not fitted	43
Total	161

Evaluation
(ODIE)

- "usefulness": useful for further investigation:

Evaluation	Number of topics
Very useful	2
Useful	12
Not useful	6

- "very useful": Query "fine"

DocID	Person	Money	Date
nyt950420	Van Halen	$1000	
nyt950704	Tarango	At least $15,500	
nyt951209	Hamilton	$12,000	this week

- "useful": Query "elect"

DocID	Position Title	Person	Date
nyt950404	president	Havel	Dec. 29, 1989
nyt950916	president	Ronald Reagan	1980
nyt951120	president	A. Kwasniewski	

Evaluation
(ODIE)

- Correctness
  evaluated 100 randomly selected rows from the "useful" and "very useful" table

Evaluation	Number of rows
Correct	84
Partially correct	4
Incorrect	12

- Role coverage: 60%

Problems and Sources of Errors

- roles are confused (e.g. victim and murderer)
- different kinds of events are found in one table
- an unrelated but collocate entity was included (“he was sentenced 3 years and fined $1,000”)

- evaluation: make the system comparable to other IE systems
- improve language analyzer components (NE, POS, corefernce resolution)
- increase portability by allowing flexible NE hierarchies

Conclusions

- Creation of tables is a clustering problem
- IE becomes a search problem:
  all tables are created preemptively, user needs to search for a relevant table.
- domain independency and possibility of n-ary relations is unique among IE systems
- “a prototype system which (...) demonstrates the feasibility of this approach”.

References

- Satoshi Sekine: On-Demand Information Extraction
- Yusuke Shinyama and Satoshi Sekine: Preemptive Information Extraction using Unrestricted Relation Discovery

Thank you for your attention!