Ontology Learning from Text

Ontology course, spring 2011
Lars Ahrenberg
May 24, 2011

Overview

- Ontologies and Natural Language Processing (NLP)
- The ontology learning layer cake
- Linguistic processing of text
- Term extraction
- Extracting synonym sets (multi-lingually)
- Learning taxonomies
- Evaluation
- Systems
- References

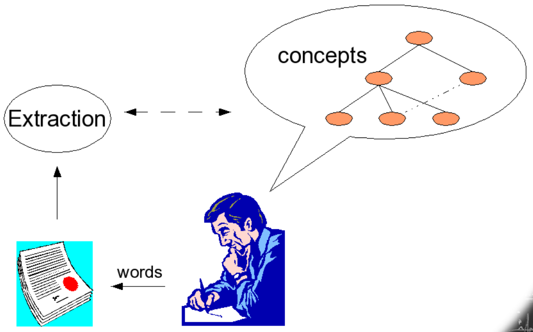
Ontologies and NLP

- Use of ontologies in NLP
  - Information extraction
  - Question answering
  - Text-to-scene mapping, ...
- Overlapping interests
  - semantics, concepts, concept relations, ...
  - Information retrieval, intelligent search, …
- Use of NLP in ontology development
  - Ontology learning
  - Ontology population

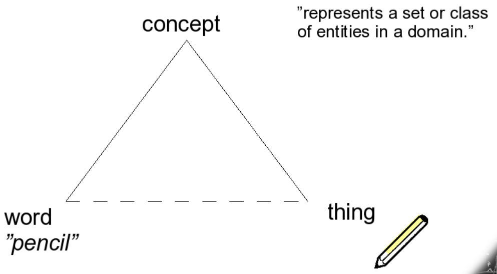
Aspects of ontology learning in the lecture

- Methods for identification of term candidates, i.e. words or phrases that are likely to express concepts.
- Methods for recognizing cross-lingual synonyms
- Methods for learning taxonomies
  - hyponyms – heteronyms (sub-concept – super-cincept)
  - co-hyponyms

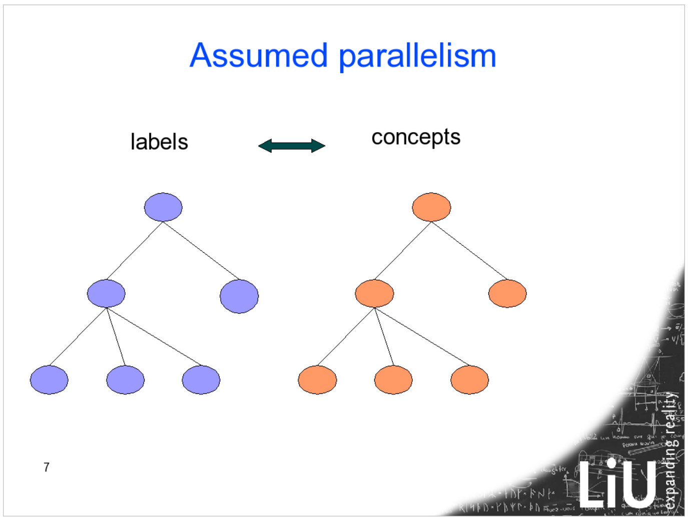
Ontology learning from text as reverse engineering



The semiotic triangle

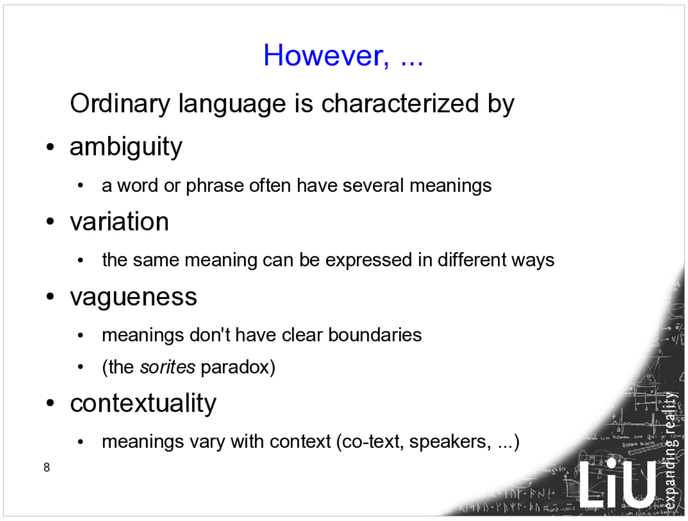


”represents a set or class of entities in a domain.”



Assumed parallelism

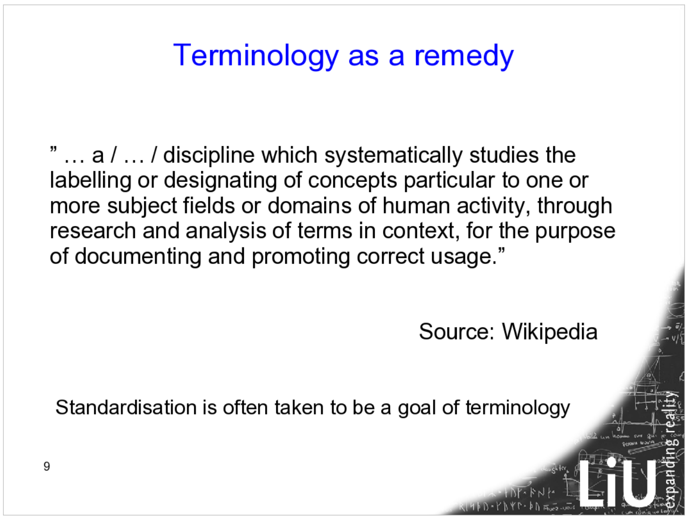
labels ⟷ concepts



However, ...

Ordinary language is characterized by

- ambiguity
  - a word or phrase often have several meanings
- variation
  - the same meaning can be expressed in different ways
- vagueness
  - meanings don't have clear boundaries
  - (the *sorites* paradox)
- contextuality
  - meanings vary with context (co-text, speakers, ...)

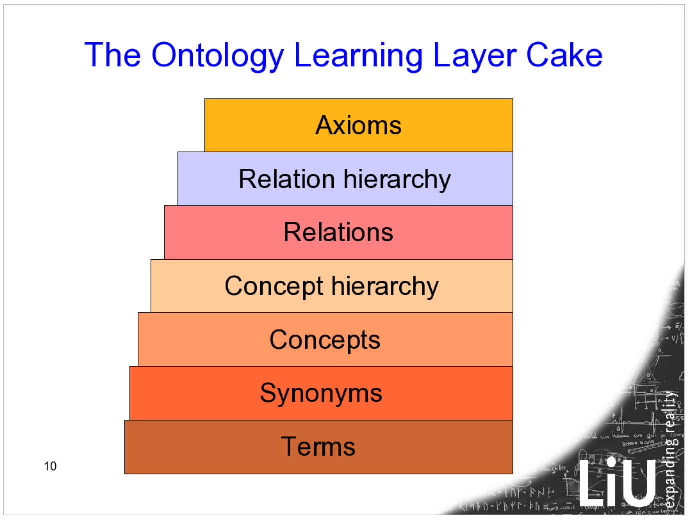


Terminology as a remedy

” … a / … / discipline which systematically studies the labelling or designating of concepts particular to one or more subject fields or domains of human activity, through research and analysis of terms in context, for the purpose of documenting and promoting correct usage.”

Source: Wikipedia

Standardisation is often taken to be a goal of terminology



The Ontology Learning Layer Cake

Axioms

Relation hierarchy

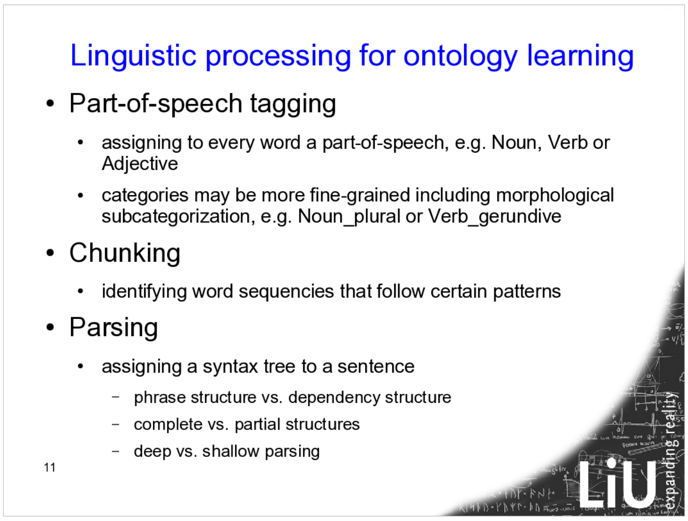
Relations

Concept hierarchy

Concepts

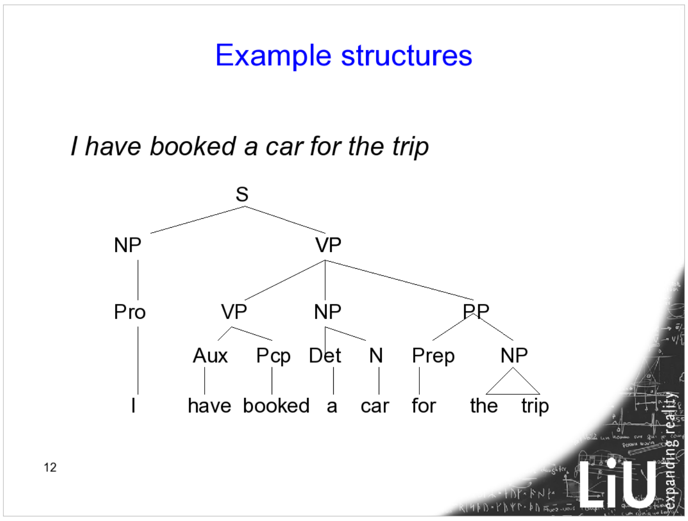
Synonyms

Terms



Linguistic processing for ontology learning

- Part-of-speech tagging
  - assigning to every word a part-of-speech, e.g. Noun, Verb or Adjective
  - categories may be more fine-grained including morphological subcategorization, e.g. Noun_plural or Verb_gerundive
- Chunking
  - identifying word sequencies that follow certain patterns
- Parsing
  - assigning a syntax tree to a sentence
    - phrase structure vs. dependency structure
    - complete vs. partial structures
    - deep vs. shallow parsing



Example structures

I have booked a car for the trip

S

NP VP

Pro VP NP PP

Aux Pcp Det N Prep NP

I have booked a car for the trip

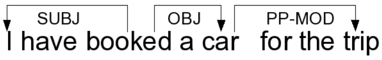
Example structures

I have booked a car for the trip

- Chunking:

  [I]$_{NP}$ [have booked]$_{VP}$ [a car]$_{NP}$ for [the trip]$_{NP}$

- Dependencies:



From dependencies to attributes

I have booked a cheap car for the trip

can yield information such as

booked_subj(I),	
booked_obj(car),	”cars are bookable”
cheap(car),	”cars can be cheap”
for_trip(car),	”cars can be used for trips”

Linguistic processing for ontology learning

- Lemmatization
  - assigning a common form to all inflectional variants
  - *book, books, book's → book_n*
  - *book, booked, books, booking → book_vb*
- Word sense disambiguation
  - identifying the sense of ambiguous words, e.g.
  - bar → bar-1, as in *chocolate bar*
  - bar → bar-2, as in *piano bar*
  - bar → bar-3, as in *standing at the bar*
  - ...

Term extraction

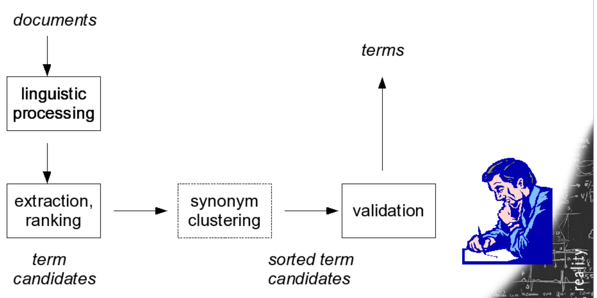
”The task here is *to find a set of relevant terms or signs for concepts and relations*, / … /, which are characteristic for the domain as represented in the underlying text collection and which will provide the basis in order to define a lexicon for an ontology … ” (Cimiano, 2010: 23, my italics)

- A problem with this definition is that it ignores (terminological) quality criteria on the lexicon
  - term or term candidate

Term extraction variants

- By text sources
  - monolingual,
  - bilingual, or
  - multilingual
- By purpose
  - ontology learning, (→ concepts)
  - translation, (→ translation lexicons)
  - standardization (→ term banks)

Term extraction, a workflow



Term candidate criteria

- Unithood
  - a term is a linguistic unit that conforms to a linguistic pattern,
  - Example (English): [Adj|Participle]* Noun* Noun (of Noun)*
  - *solution, sugar solution, supersaturated sugar solution, ...*
- Termhood (or termness)
  - a term is characteristic for a given domain
- Translatability
  - a term has synonyms in other languages

Metrics for termhood based on distribution

- distribution within documents
  - term frequency (tf)
  - occurrence by spurts
- distribution within document collection
  - tf-idf = tf * log (N/n_i) where n_i is the document frequency of the term, and N/n_i the inverse document frequency (idf)
- distribution across document collections
  - compare distribution in domain-specific collection, D, to distribution in a general text collection, G, e.g.
  - weirdness: $\dfrac{\text{count(D)*size(G)}}{\text{count(G)*size(D)}}$

Metrics for termhood

- Using statistical language models
  - the probability of a term, as estimated by a n-gram language model trained on a domain-specific collection, should be much higher than the probability given by a model trained on a general text collection.

Problems for pattern-based term extraction

- Overlap
  - *… mixed with a supersaturated sugar solution …*
  - In this sequence there are actually several candidates conforming to our part-of-speech pattern:
    - *sugar, supersaturated sugar, sugar solution,*
    - *supersaturated sugar solution*
- Modifiers that are not domain-specific
  - first, previous, following, small, limited, …
- Heads that may or may not be domain-specific
  - example, child, water, letter, piece, bit, ...

Multilingual term extraction

- Parallel corpus
  - the collection contains source documents with their translations in one or more languages.
  - sentence alignment is usually feasible
- Comparable corpus
  - the collection contains documents in different languages that are taken from the same domain
  - sentence alignment is not feasible, but document alignment might be

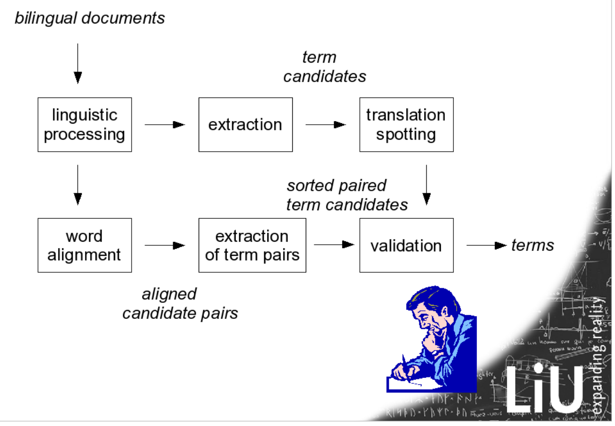
Sentence pairs of a parallel corpus

English	**Swedish**
An ANSI-89 SQL query in a database set to ANSI-92 query mode , such as :	En ANSI-89 SQL-fråga i en databas inställd i ANSI-92 frågeläge , exempelvis :
How to avoid problems caused by mixing queries under different ANSI SQL query modes in the same database	Hur du undviker problem som orsakas av att blanda frågor under olika ANSI SQL-frågelägen i samma databas
Now in Access 2002 , you can set the ANSI SQL query mode through the user interface for the current database and as the default setting for new databases .	I Access 2002 kan du ange ANSI SQL-frågeläge via användargränsnittet i den aktuella databasen och som standardinställning i nya databaser .

Alignment

- Word alignment
  - Identifying words and multi-word units that correspond under translation in a parallel corpus
  - Giza++ is a much-used tool forl word alignment based on statistical learning (Och&Ney, 2003). Provides a basis for estimating translation probabilities.
- Translation spotting
  - Given a word or multi-word unit A for one language, identify its translations in the other language(s)
  - Some simple measures of co-occurrence:
    - Dice: 2*count(A&B) / (count(A) + count(B))
    - Pointwise Mutual Information:
      log p(B|A) / p(B)

Bilingual term extraction



Semantically related words

- Synonyms
  - words that refer to the same concept
  - monolingual, e.g. *aircraft, aeroplane, plane, ...*
  - multilingual, e.g. *aircraft, flygplan, avion, Flugzeug, …*
- *Synsets*
  - sets of synonyms, representing a word sense in WordNet
- Co-hyponyms
  - words that share the same set of hypernyms
  - e.g. *apple* and *pear* are co-hyponyms under *fruit*.

Recognizing semantically related words

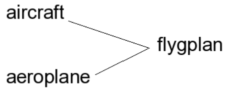
- Word clustering on distributional criteria
- **The distributional hypothesis** (Harris, 1968): Words are similar to the extent that they share contexts. The more similar the distribution, the more likely it is that they are synonyms.
- However, co-hyponyms also share contexts.
- Contexts of relevance
  - documents, words → document vectors
  - windows of, say, 2-20 words, words → context vectors
  - lexico-syntactic patterns, → word pairs with candidate relation
  - dependencies, → words and attributes in formal contexts

Recognizing synonyms

- Problem
  - Methods that rely on word distributions from monolingual douments have problems distinguishing synonyms and co-hyponyms.
- Some solutions
  - Use WordNet synsets as a resource to split clusters of related words
  - Separate multi-word terms that have different modifiers
    - supersaturated solution, non-saturated solution, solution
- Another solution
  - Bi- or multilingual data and
  - Semantic mirroring

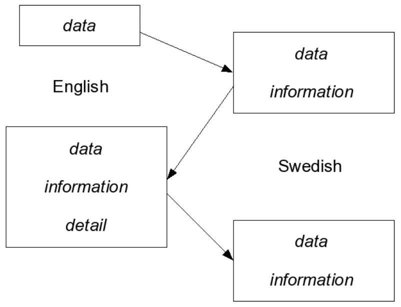
Recognizing cross-lingual synonyms

- Idea
  - If two terms are synonyms they are likely to be translated by the same term in a target language.

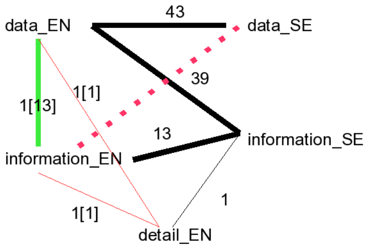


- Semantic mirroring
  - Dyvik (2002) proposed that synonyms and senses can be learnt by using translations as ”mirrors” iteratively.
  - The method can also be applied to bilingual dictionaries

Dyvik's semantic mirrors on a small corpus (LinES/Access)



A graph with the retrieved words



Bilingual term extraction and semantic mirroring

EN-SV: 1-4

EN	SV	Frekvens
assembly tool	monteringsverktyg	46
assembly tool	Monteringsvkt	3
assembly tool	monterverktyg	3
assembly tool	monteringverktyg	3

Concept clustering
EN-SV: 4-8

$ASSEMBLY TOOL-5460		
EN	**SV**	**Frekvens**
assembly tool	monteringsverktyg	46
assembly tool	Monteringsvkt	3
assembly tool	monterverktyg	3
fixture installation	verktygsmontage	1
installation tool	inställningsverktyg	4
installation tool	verktygsmontage	2
mounting tool	monteringsverktyg	1
mounting tool	uppspänningsverktyg	14
assembly tool	monteringverktyg	3
installation tool	monteringsverktyg	3

SV-EN: 1-3

EN	SV	Frekvens
assembly tool	monteringsverktyg	46
mounting tool	monteringsverktyg	1
installation tool	monteringsverktyg	3

Lexico-syntactic patterns

- Lexico-syntactic patterns (Hearst, 1992) yield hyponym and co-hyponym relations directly. Examples:
- NP1 such as NP2, NP3, and NP4
  - hyponym(lemma(head(NP2)), lemma(head(NP1)))
  - cohyponym(lemma(head(NP2)), lemma(head(NP3)))
  - *European people* such as *Swedes, Danes,* and *Norwegians*
- NP1 and other NP2
  - hyponym(lemma(head(NP1)), lemma(head(NP2)))
  - hungry *lions* and other savannah *carnivores*

Vector space methods for similarity

- When words are represented as vectors, they can be compared in a vector space.
- Similarity metrics, e.g. cosine
  - cosine(x,y) = $\frac{x \cdot y}{|x|\ |y|}$ (Note: cosine(x,x) = 1)
- Distance metrics, e.g. Euclidean distance
  - $L_2(x, y) = |x – y|$ = sqroot($\sum |x_i – y_i|^2$)
  - Note: $L_2(x,x) = 0$.

Vector elements

- The vector elements x_i, y_i, may be weights for
  - words in i context windows,
    - *eat, pick, tree, grow*, … may be context words with positive weights for various kinds of fruit and close-to-zero weights for vehicles.
  - attributes derived from dependencies,
    - eat_OBJ, grow_SUBJ, grow_OBJ, ripe_MOD, … may similarly be attributes derived from dependencies indicating fruits rather than vehicles.

Clustering in vector space

- Given a set of vectors there are several clustering algorithms that can be applied to form sets:
  - Non-hierarchical methods only form sets,
    - e.g., Kmeans that starts with k randomly chosen words/vectors and adds new ones based on distance metrics
  - Hierarchical methods, in addition, build trees
    - agglomerative clustering, where nodes represent sets, and edges represent set inclusion
      - bottom-up, starting with the terminal nodes, forming clusters based on similarity metrics

Formal concept analysis for learning hierarchies

- Formal concept analysis is a set-theoretic method for computing similarities and hierarchical relations between (formal) objects or entities.
- Objects are compared on the basis of attributes, that they may share or not share.
- The more attributes two objects share, the more similar they are.
- A set of similar objects define a local hierarchy, where the shared attributes represent a general term, or superterm.

Formal concept analysis

- A *formal context* is a triple <G, M, I> where G and M are two sets, referred to as ***objects*** and ***attributes***, and I is a binary relation on <G, M> called the ***incidence*** relation..
- Example: a tourism formal context, T (from Cimiano, 2010):

	book_OBJ	rent_OBJ	drive_OBJ	ride_OBJ	join_OBJ
hotel	X				
apartment	X	X			
car	X	X	X		
bike	X	X	X	X	
excursion	X				X
trip	X				X

Formal concept analysis

- Let O be a subset of G, A be a subset of M. Then we define
- O' = { m∈M | ∀g∈O : <g,m>∈I}

  ”the common attributes of O'
- A' = { g∈G | ∀m∈A : <g,m>∈I}

  ”the objects sharing all attributes of A”
- A pair <O,A> is a **formal concept** of <G, M, I> iff
  - O ⊆ G, A ⊆ M, O' = A, and A' = O.

Formal concept analysis

- Some formal concepts in the T example:

  O = {bike}, O' = {book_OBJ, rent_OBJ, drive_OBJ, ride_OBJ}

  O = {car, bike}, O' = {book_OBJ, rent_OBJ, drive_OBJ}

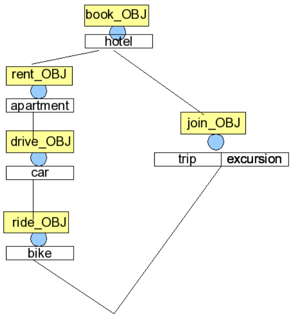
  O = {apartment, car, bike}, O' = {book_OBJ, rent_OBJ}

	book_OBJ	rent_OBJ	drive_OBJ	ride_OBJ	join_OBJ
hotel	X				
apartment	X	X			
car	X	X	X		
bike	X	X	X	X	
excursion	X				X
trip	X				X

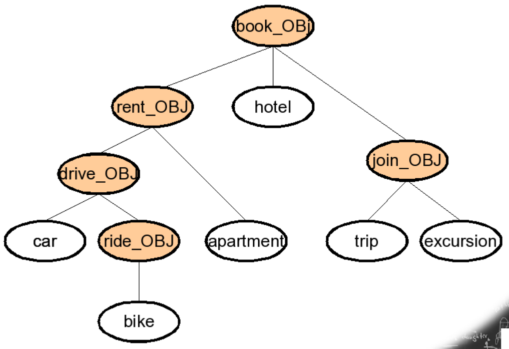
Formal concept analysis

- Formal concepts are partially ordered via the subset relation.
- **Theorem**: The formal concepts of a formal context form a complete lattice.

The lattice for the T domain visualized by reduced labelling



Forming a hierarchy using FCA



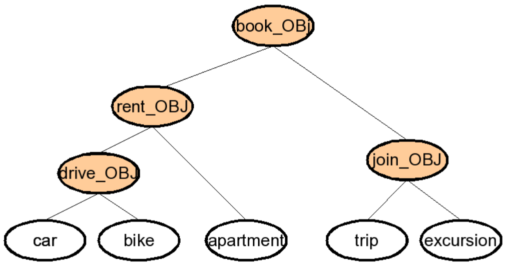
Computing similarity from attributes

- Based on occurrence:

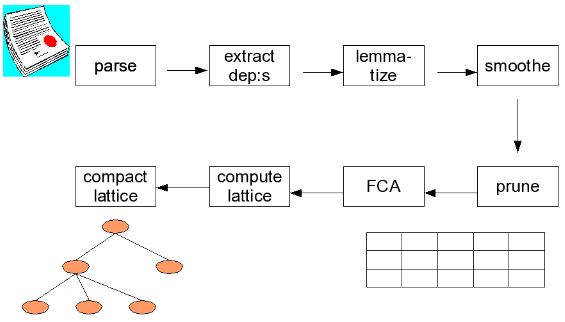
  w = shared / (shared + non-shared), threshold: 0.6

	hotel	apartment	car	bike	excursion	trip
hotel	1	0.5	0.33	0.25	0.5	0.5
apartment		1	0.67	0.5	0.33	0.33
car			1	0.75	0.25	0.25
bike				1	0.2	0.2
excursion					1	1
trip						1

Forming a hierarchy using a formal context and agglomerative clustering



A pipeline for learning hierarchical structures using formal concept analysis



Evaluation

- Precision

  P = (generated and correct) / generated
- Recall

  R = (generated and correct) / correct
- F-measure (equal weights of P and R)

  F = 2PR / (P+R)

Evaluation of term extraction

- Human validation, gives a measure of precision
  - valid term candidates, (requires a linguist)
  - validation of terms (requires domain expert)
- Comparison with gold standards,
  - both precision and recall, but gold standards are hard to construct from large data sets.
- Values vary a lot depending on corpora and resources
  - voting or combinations tend to increase values

Evaluation of taxonomies

- Human evaluation of generated pairs, i.e., by breaking up the taxonomy in its local relations and computing precision.
- Comparison with gold standards
  - Genia corpus/ontology (Tsuji labs)
  - MeSH
  - Tourism (Getess project: Staab et al. 1999; Cimiano, 2010)

Evaluation of taxonomies

- Two problems:
  - Labels may differ for the ”same” node
    - ignore labels except for terminals
    - then compute P and R as form term extraction (sometimes called 'lexical precision', 'lexical recall')
  - Measuring structural similarity
    - consider the context of a node
    - consider the ”extension” only of a non-terminal node
    - Semantic cotopy (SC), Taxonomic overlap (TO)
    - Both precision and recall (and F) may be used

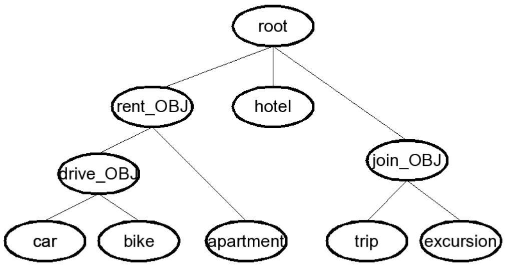
Semantic cotopy and taxonomic overlap

- $SC(c, O) := \{ c_j \text{ in } O \mid c < c_j \text{ or } c_j < c \}$
- $SC(c, O1, O2) = \{ c_j \text{ in } O1 \cap O2 \mid c < c_j \text{ or } c_j < c \}$

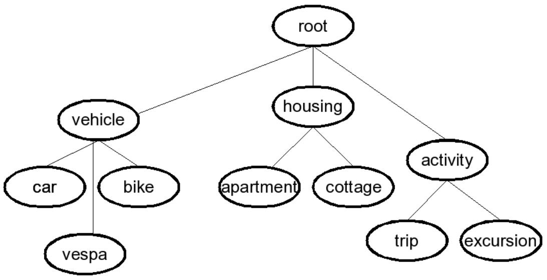
  Note! In our case, all common c:s are terminals
- $$TO(c, O1, O2) = \frac{| SC(c,O1,O2) \cap SC(c',O2,O1) |}{| SC(c,O1,O2) \cup SC(c',O2,O1) |}$$

  and c' = c (if present) or else maximally similar to c
- $$TO(O1,O2) = 1/|C| \sum_{c \text{ in } C = O1/O2} TO(c,O1,O2)$$

System output



Reference



Computations

- Lexical precision: LP = 5/6 ≈ 0.83
- Lexical recall: LR = 5/7 ≈ 0.72
- TO(join_OBJ, sys, ref) = 2/2 = 1
- TO(drive_OBJ, sys, ref) = 2/3 ≈ 0.67
- TO(rent_OBJ, sys, ref) = ¼ = 0.25
- TO(sys, ref) = 1/3(1 + 0.67 + 0.25) = 0.64
- Cimiano used a joint measure (geometric mean) of LR and F-measure from TO.

Values

- Values vary with data. Examples (from Cimiano, 2010) all best values except P_{TO} are with FCA.

Ontology	Size	Terminals	Corpus Size	Best P_{TO}	Best R_{TO}	Best F_{TO}	Best F_{LTO}
Tourism	969	796	101 M	35.2	65.5	40.5	44.7
Finance	1223	861	218 M	34.4	37.0	33.1	38.9

Some systems

Web services

- TermExtractor (http://lcl.uniroma1.it/termextractor)
- TerMine: (http://www.nactem.ac.uk/software/termine/)
- AlchemyAPI (http://www.alchemyapi.com/api/keyword/)
- Translated Labs (http://labs.translated.net/terminology-extraction)
- Yahoo http://developer.yahoo.com/search/content/V1/termExtraction.html

Downloads

- Text2Onto (http://code.google.com/p/text2onto/)

References

- Philipp Cimiano: *Ontology Learning and Population frpm Text: Algorithms, Evaluation and Applications*. Springer 2006, 2010.
- Paul Buitelaar & Philipp Cimiano. Tutorial at EACL 2006. http://people.aifb.kit.edu/pci/EACL_0L_Tutorial_06/
- Jody Foo & Magnus Merkel. Computer aided term bank creation and standardization: Building standardized term banks through automated term extraction and advanced editing tools. In Thielen & Steurs (eds.) *Terminology in Everyday Life*, John Benjamins 2010: 163-180.
- Helge Dyvik (2002). Translations as semantic mirrors: from parallel corpus to wordnet. In (eds. Karin Aijmer and Bengt Altenberg) *Language and Computers, Advances in Corpus Linguistics. Papers from the 23rd International Conference on English Language Research on Computerized Corpora* (ICAME 23) Göteborg 22-26 May 2002. pp. 311-326

Identifying other relations

- Lexico-syntactic relations
  - might be many
- Bootstrapping
  - seed patterns
  - Example:
    - /varslar [0-9]+ /
    - /Volvo [a-zåäö]+ [a-zåäö0-9|\b]{0,3} anställda/

Att använda mönsterfrön

Seed pattern?

- [COMPANY] varslar [NO.] [EMPLOYEES]

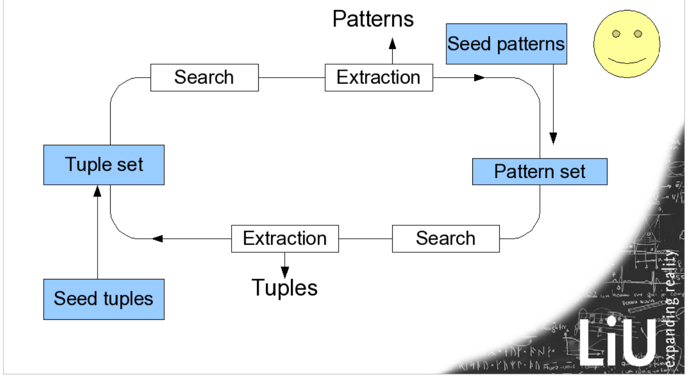
  yields tuples, e.g.

  <Volvo, 100, anställda>

  <Saab, 400, ingenjörer>

  …

Boot-strapping



Patterns

Seed patterns

Search

Extraction

Tuple set

Pattern set

Extraction

Search

Tuples

Seed tuples