From linguistic predicate-arguments to Linked Data and ontologies

Extracting n-ary relations

Stamatia Dasiopoulou
Gerard Casamayor
Simon Mille

Universitat Pompeu Fabra, Barcelona, Spain

Heraklion Greece, May 29th 2016
Tutorial Objectives

• Introduction to Natural Language Processing tools and resources for predicate-argument identification in text

• Overview of models and methods for mapping linguistic structures to SW representations
Tutorial Structure

• **9:00-9:20** Part I - Introduction
• **9:20-10:30** Part II - From text to semantic structures
• **10:30-11:00** Coffee break
• **11:00-12:30** Part III - From linguistic representations to RDF/OWL
• **12:30-14:00** Lunch break
• **14:00-14:45** Part IV – Example applications & Evaluation methods
• **14:45-15:30** Hands-on session (I)
• **15:30-16:00** Coffee break
• **16:00-17:30** Hands-on session (II)
Part I: *From linguistic predicate-arguments to LD and ontologies – Extracting n-ary relations*

Introduction
Knowledge Extraction from NL texts for the Semantic Web

• **Goal:** capture information in NL texts and model it in a SW compliant form to allow further processing

• What further processing?
  – Ontology learning and population
  – Natural Language Interfaces
    • Question Answering over KBs
    • NL dialogue systems
  – Semantic search & retrieval
  – Natural Language Generation from ontologies & KBs
    • summarisation, ontology verbalizers, etc.
Typical KE tasks

- **Named Entity Recognition (NER)**
  - Detects NEs, often referred to with proper names: people, places, organizations, etc.
  - Tools: Stanford NER, Illinois NET, OpenCalais NER, etc.

- **Entity Linking (EL)**
  - Identity resolution by linking to external knowledge resources
    - Resources: WordNet, Wikipedia, DBpedia, BabelNet, etc.
  - Tools: Babelfy, DBpedia Spotlight, AGDISTIS, NERD, etc.

- **Binary relation extraction**
  - Corresponds to the typical <subject, predicate, object> triple
  - Tools: REVERB, DBpedia population tools, etc.

- **N-ary relation extraction**
  - Complex relational contexts (events, situations) involving multiple entities
  - Tools: FRED, PIKES, taInSRL, etc.
NLP tasks relevant to KE

• Coreference resolution
  – Detects text fragments referring to the same entity
  – Tools: Stanford CoreNLP, Berkeley Coreference Resolution System, etc

• Word Sense Disambiguation (WSD)
  – Assigns senses to words according to some dictionary of senses.
  – Resources: WordNet, BabelNet, etc.
  – Tools: UKB, Babelfy, etc.

• Syntactic parsing
  – Captures the functional structure of sentences
  – Resources: Penn Treebank, CCGBank, etc.
  – Tools: Mate tools, Stanford CoreNLP, Berkeley tools, etc.

• Semantic Role Labelling (SRL)
  – Assigns semantic types to predicate-argument structures
  – Resources: PropBank, NomBank, VerbNet, FrameNet, etc.
  – Tools: Semafor, Mate tools, etc.

• Semantic Parsing
  – Produces (linguistic) semantic structures of sentences of whole texts
  – Resources: DRS, ARM,
  – Tools: Boxer, ARM parsers, talnSLR
N-ary relations

Christine has breast tumour with high probability.

Steve has temperature, which high, but falling.

John buys a "Lenny the Lion" book from books.example.com for $15 as a birthday gift.

United Airlines flight 3177 visits the following airports: LAX, DFW, and JFK.

https://www.w3.org/TR/swbp-n-aryRelations/
N-ary relations

• **Goal:** Capture the n-ary dependencies!
  – e.g. “John gave a book to Mary on Wednesday.”
    “John gave a book to Alice on Thursday.”
    “John gave a pen to Paul on Wednesday.”

• Query data precisely!
  – What did John give to Mary?
  – To whom did John give a book?
  – What did John on Wednesday?
  – What events/situations did John instigate?
  – What events/situations involve John and Mary?
N-ary relation extraction

• How to go from text to an ontology
  – Many sentences with the same meaning!
  – Conceptual gap!

  - John gave Mary a book.
  - John gave a book to Mary.
  - Mary was given a book by John.
  - What John gave to Mary was a book.
  - John a donné un livre à Mary.
N-ary relation extraction

• Easier to get there in several steps
  – A linguistic structure that is close to the text – structure over all the words and only these words
    ➢ *Syntactic structure*

  – A linguistic structure that abstracts from language-specific features and is closer to the meaning
    ➢ *Semantic structure*
Predicates & Arguments

- **Predicate-argument** structures are the basis of many semantic structures
  - e.g. Discourse Representation Theory (DRT), Abstract Meaning Representation (ARM), SemS in Meaning-Text Theory

- A **predicate** is a linguistic element whose meaning isn’t complete without its arguments
  - e.g. *yellow* submarine
  - e.g. John *bought* a book from Mary for 5€
Predicates & Arguments

• Predicate-argument structures can be derived from the output of syntactic parsers

• However:
  – Parsers do not assign semantic types
  – May not identify semantic relations not expressed syntactically
  – Functional words need to be removed
  – Do not determine the specific entities or concepts participating in the relation

• Other tools are needed to produce semantic structures:
  – SRL, semantic parsers, NER, EL, WSD, etc.
Semantic analysis

- **Semantic Role Labelling (SRL) tools** type predicate-argument relations
  - Semantic role resources: NomBank, PropBank, VerbNet, FrameNet, etc.
  - E.g. Semafor, Mate Tools SRL, Illinois SRL, etc.

- **Semantic parsers** produce semantic **structures** for sentences or whole texts
  - Tools: Boxer (DRS), JAMR (AMR), etc.

- Semantic role **resources**: NomBank, PropBank, VerbNet, FrameNet, etc.

- E.g. “John **bought** a book from Mary for 5€”

<table>
<thead>
<tr>
<th>SRL resource</th>
<th>John</th>
<th>buy</th>
<th>book</th>
<th>Mary</th>
<th>5€</th>
</tr>
</thead>
<tbody>
<tr>
<td>PropBank</td>
<td>A0</td>
<td>buy.01</td>
<td>A1</td>
<td>A2</td>
<td>A3</td>
</tr>
<tr>
<td>VerbNet</td>
<td>Agent</td>
<td>get-13.5.1</td>
<td>Theme</td>
<td>Source</td>
<td>Asset</td>
</tr>
<tr>
<td>FrameNet</td>
<td>Buyer</td>
<td>Commerce_buy</td>
<td>Goods</td>
<td>Seller</td>
<td>Money</td>
</tr>
</tbody>
</table>
Bridging NLP and SW

- Part II – Abstracting text
  - Coverage of linguistic phenomena
  - Richness and expressiveness of semantic structures and types

- Part III – From linguistic representations to SW
  - Still a large gap between extracted NLP structures and formal semantics in the SW
    - Relations modelled using RDF versions of lexical resources
    - Non-trivial design & re-engineering is required
Part II – From text to linguistic semantic structures
Outline

• Syntactic parsing
  – Constituency parsing
  – Dependency parsing

• Semantic resources and tools
  – SRL
    • ProbBank/NomBank
    • VerbNet
    • FrameNet
  – Semantic parsing:
    • DRS
    • AMR
    • Semantic Dependencies

• From syntax to semantics
• Understanding words is obviously not enough in order to understand a text
• Need for structuring natural language in order to access the meaning of a sentence
  – More or less abstract ways
  – Syntactic parsing VS Semantic Role Labeling
Syntactic parsing

- Objective: capture the **sentence** structure!
  - how **words** relate together
  - specific to each particular language
  - 2 main paradigms:
    a. constituency-based models: grouping
    b. dependency-based models: linking

(a) [[[The dog] [chases [the mouse]]]]
(b) The dog chases the mouse
Constituency Parsing (I)

- Describes phrase groupings (constituents)
  - Based on word order
- Non-terminal elements are **constituent** names
  - Noun Phrase, Verb Phrase, etc.
  - Eg: a well-formed NP is:
    - a sequence of determiner and a noun
    - a sequence of a determiner, an adjective and a noun, etc.

```
[ [The dog]_{NP} [chases [the mouse]_{NP} ]_{VP} ]_{Sent}
```
Constituency Parsing (II)

- A constituency parser:
  - automatically produces word groupings
  - predicts if a sentence is grammatically correct or not

- has problems with free-order languages
- Predicate-argument relations can be mapped indirectly
- Automatic derivation of approximate dependencies
Dependency Parsing (I)

• Describes syntactic relations between words (dependencies)
  – Independent of word order

• Edge labels indicate grammatical function of a dependent to a governor:
  – Encode agreements, ordering, functional word requirements
  – E.g.: a subject is typically an element that triggers number and person agreement on the verb and that is realized before it in a neutral sentence.
Dependency Parsing (II)

• A dependency parser:
  – automatically produces dependencies between words
  – does not predict if a sentence is grammatically correct or not
    • ... but can model any sentence
  – has no problem with free-order languages
  – Straightforward mapping to predicate-argument relations
  – Automatic derivation of approximate constituency structure
Dependency Parsing (III)

- Many different annotation philosophies
- Different perspectives regarding:
  - notion of governor/dependent
  - edge label nomenclature

Penn Treebank style
(Johansson & Nugues, 2007)

USD style
(De Marneffe & Manning, 2008)
Parsing Pipelines

- Typical parsing pipeline
  - Input
  - Tokenization
  - Lemmatization
  - Part-of-Speech tagging
  - Dependency labelling
  - Constituency labelling

  - won’t
  - wo + n’t
  - will + not
  - willₐ + notₐ
  - willₐ –ADV-> notₐ
  - [will not]ₐ
Tools for parsing

• Rule-based parsers
  – not always reliable (coverage issues)
  – output depends on the rules

• Statistical parsers
  – parser + model trained on annotated data
  – output depends on the training data
    • one parser can work with different models
  – always return a complete structure
A few popular parsers

• Dependency (Statistical)
  – Stack LSTM (new): neural networks, state-of-the-art
    • (Dyer et al., 2015)
  – MATE-Tools (PTB)
    • (Björkelund et al., 2010) http://en.sempar.ims.uni-stuttgart.de/
  – TurboParser (USD)
    • (Martins et al., 2010) http://www.cs.cmu.edu/~ark/TurboParser/

• Constituency (Statistical and Rule-based)
  – CCG parsers
    • (Clark & Curran, 2007) http://4.easy-ccg.appspot.com/
  – Stanford parser
    • (Klein & Manning, 2003) http://nlp.stanford.edu:8080/parser/index.jsp
Towards semantic parsing (I)

• Constituency parsing tells us that:
  – “the mouse” and “the dog” are noun groups
  – “chases the mouse” is a verb group
  – “the dog chases the mouse” is a complete sentence

• Dependency parsing tells us that:
  – “dog” is subject of “chases”
  – “mouse” is object of “chases”

• How to go towards linguistic meaning?
Towards semantic parsing (II)

• **Objective:** get closer to the *meaning*
  – abstract away from language-specific features
    • syntactic relations (constituencies, dependencies)
    • functional words
      – determiners, void conjunctions/prepositions, auxiliaries, etc.
  – capture relations not expressed in syntax
  – common use of predicate-argument structures
    • meaningful words + formalized semantic relations
Towards semantic parsing (III)

- Lexical resources for typing of semantic-predicate structures:
  - PropBank (Kingsbury & Palmer, 2002) / Nombank (Meyers et al., 2004)
  - VerbNet (Schuler, 2005)
  - FrameNet (Baker et al., 1988)

- Formalisms for semantic structures:
  - Discourse Representation Structures (DRS, Kamp et al., 2011)
  - Abstract Meaning Representation (AMR, Banarescu et al., 2012)
  - Others: Meaning-Text Theory SemS, Prague T-Layer, Minimal Recursion Semantics, Enju, Collapsed Stanford, etc.
Outline

• Syntactic parsing
  – Constituency parsing
  – Dependency parsing

• Semantic resources and tools
  – SRL
    • ProbBank/NomBank
    • VerbNet
    • FrameNet
  – Semantic parsing:
    • DRS
    • AMR
    • Semantic Dependencies

• From syntax to semantics
PropBank/NomBank (I)

- 2 English lexicons [http://verbs.colorado.edu/verb-index/](http://verbs.colorado.edu/verb-index/)

**Contents**

- about 11,000 **predicative words** with one or more **senses**:
  - e.g., senses for verb chase: chase.01, chase.02
- for each sense, arguments are assigned **semantic roles**:
  - Universal set of predicate-argument roles: A0, A1, A2, A3, A4, A5
  - +AM-.., R-A.., C-A..

**Corpus**

- Penn Treebank syntactic dependencies annotated with PropBank/NomBank
PropBank/NomBank (III)

<?xml version="1.0"?>
<!DOCTYPE frameset SYSTEM "frameset.dtd">
<frameset>
  <predicate lemma="chase">
    <note> Frames file for 'chase' based on sentences in financial subcorpus. Verbnet entry chase-51.6, other framed members include 'follow' and 'pursue'. Comparison with follow. </note>
    <roleset vncls="51.6" name="follow, pursue" id="chase.01">
      <roles>
        <role n="0" descr="follower">
          <vnrole vncls="51.6" vnttheta="Agent"/>
        </role>
        <role n="1" descr="thing followed">
          <vnrole vncls="51.6" vnttheta="Theme"/>
        </role>
      </roles>
      <example name="with causal agent">
        <text> John chased his dinner with a shot of antifreeze. </text>
        <arg n="A">John</arg>
        <rel>chased</rel>
        <arg n="1">his dinner</arg>
        <arg n="M" f="MNR">a shot of antifreeze</arg>
      </example>
      <example name="with directional">
        <text> Higher margins would chase away dozens of smaller traders who help larger traders buy and sell, they say. </text>
        <arg n="0">Higher margins</arg>
        <arg n="M" f="MOD">would</arg>
        <rel>chase</rel>
        <arg n="M" f="DIR">away</arg>
        <arg n="1">dozens of smaller traders who help larger traders buy and sell</arg>
      </example>
    </roleset>
  </predicate>
</frameset>
PropBank/NomBank (IV)

- Sample tool: Mate tools SRL
  - [http://en.sempar.ims.uni-stuttgart.de/parse](http://en.sempar.ims.uni-stuttgart.de/parse)

- **PropBank**: Senses are specific to each linguistic predicate
  - no common semantic types for synonymous or quasi-synonymous verb senses
    - e.g. buy.01 and purchase.01

- **Mate tools**: Predicate-argument structures can contain syntactic information
  - relative clauses, circumstantials
  - no distinction between semantic and functional prepositions

- **Mate tools**: Structures can be incomplete
  - disconnections (partial coverage of all PoS)
  - annotates arguments as spans
VerbNet (I)

- Theory: Levin verb classes
- English lexicon [http://verbs.colorado.edu/verb-index/](http://verbs.colorado.edu/verb-index/)
- Contents
  - 270 semantic classes of verbs inspired from Levin
    - e.g.: Putting, Removing, etc.
    - each class has a list of members (PropBank entries)
  - each class is described with a set of arguments and a set of syntactic frames:
    - about 40 universal thematic roles: Agent, Patient, Beneficiary, etc.
    - selectional restrictions: concrete, abstract, location, organization, etc.
    - correspondence with generic predicate-argument structure
- Corpus
  - collection of examples that illustrate VerbNet entries
VerbNet (III)

- `<FRAMES>`
  - `<FRAME>`
    - `<DESCRIPTION xtag="0.2" secondary="Basic Transitive" primary="NP V NP" descriptionNumber="0.2"/>`
  - `<EXAMPLES>`
    - `<EXAMPLE>``Jackie chased the thief.</EXAMPLE>`
  - `<EXAMPLES>`
    - `<SYNTAX>`
      - `<NP value="Agent">`
        - `<SYNRESTRS/>`
      - `<NP value="Theme">`
        - `<SYNRESTRS/>`
    - `<NP>`
      - `<VERB/>`
    - `<NP value="Agent">`
      - `<SYNRESTRS/>`
    </NP>`
  </SYNTAX>`
- `<SEMANTS>`
  - `<PRED value="motion">`
    - `<ARGS>`
      - `<ARG type="Event" value="during(E)"/>
        - `<ARG type="ThemRole" value="Agent"/>
      </ARGS>`
    - `<PRED>`
    - `<PRED value="motion">`
      - `<ARGS>`
        - `<ARG type="Event" value="during(E)"/>
          - `<ARG type="ThemRole" value="Theme"/>
        </ARGS>`
      - `<PRED>`
    </SEMANTS>`
  - `<FRAME>`
    - `<DESCRIPTION xtag="" secondary="np-PPpath-PP" primary="NP V NP PP.location" descriptionNumber="0.2"/>`
  - `<EXAMPLES>`
    - `<EXAMPLE>``Jackie chased the thief down the street.</EXAMPLE>`
  - `<EXAMPLES>`
VerbNet (IV)

• Demo: see DRS (next)
• Very precise description of participants
  – semantics + syntax
• Easily reachable from predicate-argument structures produced by syntactic structures
• Semantic classes abstract over language-specific predicates
  • !! VerbNet: Classes are coarse-grained
  • !! VerbNet: Unstructured set of classes, without semantic relations between them.
FrameNet (I)

- Linguistic theory: Frame Semantics (Fillmore)
- English lexicon [https://framenet.icsi.berkeley.edu/fndrupal/frameIndex](https://framenet.icsi.berkeley.edu/fndrupal/frameIndex)
- Contents
  - 1,000 + **semantic frames**
    - frames describe types of situations, objects, or events
      - e.g. Grooming, Absorb_heat, Usefulness, Food, etc.
    - and are associated with Lexical Units, i.e. words, that evoke them
      - e.g. Grooming LUs include: bathe, shower, comb, floss, shave, etc.
  - for each frame, a set of frame-specific roles is defined
    - about 330 **frame elements**: Buyer.commerce_buy, Ingestor.ingestion, etc.
- Corpus
  - collection of sentences that illustrate FrameNet entries
FrameNet (II)

**Cotheme**

**Definition:**

This frame contains words that necessarily indicate the motion of two distinct objects. The Theme is typically animate and is expressed the same way a Self-mover is expressed in the Self-motion frame—i.e., as the subject of a target verb. The Cotheme may or may not be animate and is typically expressed as a direct object or an oblique. Source, Path, Goal, and the other frame elements common to motion words also regularly occur with the words in this frame.

**Lexical Unit Index**

<table>
<thead>
<tr>
<th>She</th>
<th>ACCOMPANIED</th>
<th>him</th>
<th>in</th>
<th>the</th>
<th>ambulance</th>
<th>to</th>
<th>Hollywood</th>
<th>Memorial</th>
<th>Hospital</th>
</tr>
</thead>
<tbody>
<tr>
<td>Police</td>
<td>PERSUE</td>
<td>teenage</td>
<td>joyrider</td>
<td>across</td>
<td>three</td>
<td>counties</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>My</td>
<td>PURSUIT</td>
<td>of</td>
<td>academic</td>
<td>superiority</td>
<td>shows</td>
<td>that</td>
<td>I</td>
<td>was</td>
<td>still</td>
</tr>
</tbody>
</table>

*Relation Extraction - ESWC’16 Tutorial*
### FrameNet (III)

**Area**
- **Semantic Type:** Location
- **Examples:** Area marks expressions which describe a general area in which the motion of Theme and Cothema takes place when the motion is understood to be irregular and not to consist of a single linear path. Locative setting adjuncts of motion expressions may also be assigned this frame element.
  - The police **followed** the suspects **all around town**.

**Cothema**
- **Semantic Type:** Physical_object
- **Examples:** This frame element is the second moving object, expressed as a direct object or an oblique.
  - Pat **accompanied** me down the street.
  - The squirrel **chased** after the nut.

**Direction**
- **Semantic Type:** dir
- **Examples:** The direction in which the Theme and Cothema move.
  - The tracking dogs **chased** their target towards the river, where they lost it.

**Goal**
- **Semantic Type:** Goal
- **Examples:** Any expression which tells where the Cothema ends up as a result of the motion expresses the frame element Goal. Note that if the Cothema is animate, the Theme need not also end up in the same place. Some particles imply the existence of a Goal which is understood in the context of utterance.
  - The children **chased** the ball **into the park** (The children end up in the park.)
  - The children **chased** the dog **into the park** (The dog ends up in the park; the children may not have entered the park.)

**Path**
- **Semantic Type:** Path
- **Examples:** Path marks phrases that describe the Theme and Cothema's trajectory of motion and which are neither expressions of the Source nor the Goal of motion. The motion Path also includes directional expressions.
  - The bikers **followed** the truck **through the desert**.

**Read**
- **Semantic Type:** Road
- **Examples:** Phrases that denote a physical path on which the motion of Theme and Cothema takes place are marked Read.
  - The police **followed** the suspect **on the interstate for several miles**.

**Source**
- **Semantic Type:** Source
- **Examples:** Source marks any expression which implies a definite starting-point of the motion of the Cothema. In prepositional phrases, the object expresses the starting point of motion. With particles, the starting point of motion is understood from context.
  - The cat **chased** the mouse **out of the house**.

**Theme**
- **Semantic Type:** Physical_object
- **Examples:** This is the entity, frequently a living being, which moves in relation to the Cothema. Normally the Theme frame element is expressed as an external argument.
  - Pat **accompanied** me for five miles in a blue Toyota.
  - The squirrel **chased** the nut across the road.
FrameNet (IV)

- Sample tool: Semafor
  - [http://demo.ark.cs.cmu.edu/parse](http://demo.ark.cs.cmu.edu/parse)

- Frames abstract over language-specific predicates
- Frame to Frame relations
  - inheritance*, subframe, using (perspectivises), causativeOf & inchoactiveOf, precedence, and seeAlso
- Corpus-based descriptions of syntactic frames
- **FrameNet**: Mapping from predicate-arguments to roles is complex
- **Semafor**: structures can be incomplete
  - disconnections (partial coverage of all PoS)
  - annotates arguments as spans
Linking between PropBank/NomBank, VerbNet and FrameNet

• SemLink
  – (Palmer, 2009)
  – PropBank-VerbNet
  – VerbNet-FrameNet

• PredicateMatrix
  – (De Lacalle et al., 2014)
  – PropBank, VerbNet, FrameNet, WordNet, Multilingual Central Repository
Outline

• Syntactic parsing
  – Constituency parsing
  – Dependency parsing

• Semantic resources and tools
  – SRL
    • ProbBank/NomBank
    • VerbNet
    • FrameNet
  – Semantic parsing:
    • DRS
    • AMR
    • Semantic Dependencies

• From syntax to semantics
Discourse Representation Structures (I)

• Based on Discourse Representation Theory (DRT)
  – First Order Logic formal theory of meaning

• Phenomena covered:
  – Predicate-argument relations
  – Negation, equivalence, quantifiers, sets, modality, etc.

• DRSs building blocks
  – Discourse referents, i.e. variables corresponding to discourse entities (event, objects, etc.)
  – Conditions, i.e. constraints (properties, relations) on discourse entities: e.g. chase(x, y), dog(x), mouse(y)

• Corpus
  – Groningen Meaning Bank: http://gmb.let.rug.nl/
  – Whole texts are annotated rather than isolated sentences
Discourse Representation Structure (II)

- DRS semantic parser: **Boxer**
  - [http://gmb.let.rug.nl/webdemo/demo.php#drs](http://gmb.let.rug.nl/webdemo/demo.php#drs)
  - Event-based Neo-Davidsonian representation (reified n-ary relations) using VerbNet thematic roles
  - e.g. “The dog wants to chase the mouse.”

- FOL representation

$$\exists \ A : \exists \ B : \ ( ( \ n1\text{mouse}(A) \land n1\text{dog}(B) ) \land \exists \ C : \exists \ D : \ ( \ r1\text{Topic}(C, D) \land ( \ r1\text{Actor}(C, B) \land ( \ v1\text{want}(C) \land \exists \ E : ( \ r1\text{Theme}(E, A) \land ( \ r1\text{Actor}(E, B) \land v1\text{chase}(E) ) ) ) ) ) )$$
Abstract Meaning Representation (I)

• Abstract Meaning Representation (AMR)
  [http://amr.isi.edu/language.html](http://amr.isi.edu/language.html)

• Phenomena covered:
  – Predicate-argument relations
  – Within-sentence coreference
  – NEs and types
  – Modality, negation, questions, quantities, etc.

• AMRs are rooted, labelled graphs
• Makes extensive use of PropBank
• Corpus: AMR Bank containing English sentences.
Abstract Meaning Representation (II)

- **AMR concepts** English words (boy), PropBank framesets (want-01), special keywords.

- **Keywords** entity types (date-entity, world-region), quantities (monetary-quantity, distance-quantity), and logical conjunctions (and).

- Probank **arguments** arg0, arg1, etc.

- Universal set of **semantic relations**: accompanier, age, beneficiary, cause, etc.

**LOGIC format:**

\[ \exists w, b, g: \\
\text{instance}(w, \text{want-01}) \land \text{instance}(g, \text{go-01}) \land \\
\text{instance}(b, \text{boy}) \land \text{arg0}(w, b) \land \\
\text{arg1}(w, g) \land \text{arg0}(g, b) \]

**AMR format** (based on PENMAN):

\[
(w / \text{want-01} \\
: \text{arg0} (b / \text{boy}) \\
: \text{arg1} (g / \text{go-01} \\
: \text{arg0} (g, b))
\]

**GRAPH format:**

```
instance
want-01
ARG0
ARG1
ARG0
instance
boy
```

```
go-01
```
Abstract Meaning Representation (III)

- AMR semantic parser: JAMR
  [https://github.com/jflanigan/jamr](https://github.com/jflanigan/jamr)

- **AMR**: Annotation at sentence-level only (no discourse).
- **AMR**: Uses propbank
- **AMR**: Heavily geared towards English
- **JAMR**: English only
Semantic Dependencies (I)

- Semantic dependencies
- Similar formalism to syntactic dependencies, but with word-to-word semantic relations between content words.
- Phenomena covered:
  - Predicate-arguments relations.
  - Modality, appositions, argument sharing in coordinations and complex verb constructions, etc.
- Can integrate with PropBank, VerbNet or FrameNet.
Semantic Dependencies (II)

- Semantic dependencies parser: talnSRL
  [http://dpars.e.mltisensor.taln.upf.edu/demoMS/relExtr/](http://dpars.e.mltisensor.taln.upf.edu/demoMS/relExtr/)

- Multilingual analysis (in progress).

- NER and EL

- **!! Dependencies**: Annotation at sentence-level only (no discourse).

- **!! Dependencies**: No semantic interpretation of negations, sets and quantifiers (yet).
Outline

• Syntactic parsing
  – Constituency parsing
  – Dependency parsing

• Semantic resources and tools
  – SRL
    • ProbBank/NomBank
    • VerbNet
    • FrameNet
  – Semantic parsing:
    • DRS
    • AMR
    • Semantic Dependencies

• From syntax to semantics
From dependencies to semantics (I)

- Parallelism between consecutive outputs (taInSRL)

1. surface-syntax

2. deep-syntax

3. PropBank/NomBank

4. VerbNet

5. FrameNet
From dependencies to semantics (II)

• What has to be done starting from syntactic dependencies:
  – Identify functional nodes to remove
  – Find new governor/dependent pairs
  – Label the edges

• ML (using annotated corpora) or rules (using SemLink)

• Mapping not always one-to-one
  – e.g. SBJ-> I/II if active/passive
  – Even more complex examples!
From constituents to semantics (I)

- Example: CCG to DRS
  - The mapping is not straightforward
  - Semantic types (lambda expressions) can be manually assigned to words and to constituents
From constituents to semantics (II)
From constituents to semantics (III)

• What has to be done starting from constituency structures:
  – ensure that there is a correct type corresponding to each constituent
  – ensure that they can combine properly

• Lambda-expressions must pre-exist
  – !! potential coverage issues
From syntax to semantics

• Disambiguation needed when the predicate has multiple senses/VerbNet classes/frames
  – e.g. play: play.01, play.02?

• Semantic types of arguments can help to determine roles and resolve selectional restrictions
  – Play football with Peter
  – Play Peter with football
Summary (I)

• Variety of linguistic types:
  – Predicates: disambiguated words, VerbNet classes, FrameNet frames
  – Argument roles: A0, Agent, Buyer.commerce_buy...
  – Structures: DRS, AMR, semantic dependencies, etc.

• Both constituency and dependency parsing can be used to extract these representations

• Linguistic resources are crucial!
  – Linguistic theories -> formal definitions of structures
  – Annotated corpora -> to train models for tools
  – Semantic lexicons -> to assign semantic types
  – Rule-sets -> to implement parsers or mappings between structures
Summary (II)

- Semantic parsers and SRL tools work naturally with n-ary relations!
Summary (III)

• What semantic representation is best?
  – Depends on the application:
    • For knowledge extraction, disconnected representations may be ok, e.g. predicate-argument structures identified by SLR tools.
    • For summarization, comprehensive structures covering whole sentences are needed!
  – In general, the more expressive and accurate the abstraction the better starting point it serves

• How to formalize linguistic structures to Semantic Web compliant ones?
  – Part III
Part III – From linguistic representations to RDF/OWL
Outline

• Porting linguistic data to RDF
  – Models
    • NIF, EARMARK
    • PROV-O, ITS and NERD, OLiA
    • lemon
    • PreMOn

• Mapping linguistic structures to SW representations
  – Some relevant background
    • DUL DnS, c.DnS, LMM
  – Approaches
    • LODifier
    • taInSRL
    • PIKES
    • FRED
Bridging NL and the SW

1. Port linguistic resources and output of NLP tools to RDF
   – Motivated by reuse & interoperability of linguistic resources and tools
   – Publish Linguistic Open Linked Data

2. Translate linguistic structures into SW knowledge representations
   – Distil conveyed knowledge
   – Allow for automated reasoning
The Linguistic LOD cloud (LLOD)

http://linguistic-lod.org/
Porting NLP to the SW

• Relevant models for publishing LLOD

  1. Ontologies used to model the textual annotations produced by NLP and KE tools
     • e.g. NIF, EARMARK, PROV-O, OLiA, ITS, NERD

  2. Ontologies used to associate linguistic information to existing ontologies and LD
     • e.g. lemon, OntoLex

  3. Existing Linked Data versions of linguistic resources, e.g. lexica and corpora
     • e.g. PreMOn, WordNet RDF, FrameNet RDF, etc.
NIF

• Set of specifications, ontologies and software to achieve interoperability between NLP tools, resources and annotations.

• Core ontology for modelling of stand-off annotations of text.

• Used by TALN-UPF SRL, FRED and PIKES.

http://persistence.uni-leipzig.org/nlp2rdf/

<http://brown.nlp2rdf.org/corpus/a01.xml#offset_4_10>  
a nif:String , nif:Word , nif:OffsetBasedString ;  
nif:anchorOf "Fulton"^^xsd:string ;  
nif:referenceContext <http://brown.nlp2rdf.org/corpus/a01.xml#offset_0_161> ;  
nif:beginIndex "4"^^xsd:int ;  
nif:endIndex "10"^^xsd:int ;  
a nif:EntityMention ;  
nif:confidence “0.98”^^xsd:decimal .
EARMARK

• Ontology for modelling non-embedded markup on documents, such as XML and XHTML.

• It has also been used to model stand-off semantic annotations of text in FRED.

```xml
# The textual content of the document to annotate
:content a earmark:URIDocuverse;
  earmark:hasContent
  "http://ota.ox.ac.uk/text/5725.xml"^^xsd:anyURI.
# The string "Ari."
:ari-string a earmark:PointerRange;
  earmark:refersTo :content;
  earmark:begins "33974"^^xsd:nonNegativeInteger;
  earmark:ends "33978"^^xsd:nonNegativeInteger.
```
PROV-O

• Ontology of the PROV set of documents to model provenance information.
• Extended to model provenance of NLP and IE annotations in NIF 2.0 and 2.1 draft.

https://www.w3.org/TR/prov-o/

```xml
<http://brown.nlp2rdf.org/corpus/a01.xml#offset_4_10>
a nif:Word ;
nif:anchorOf "Fulton"^^xsd:string ;
nif:annotationUnit [ 
a nif:EntityMention ;
nif:confidence "0.98"^^xsd:decimal
prov:wasAttributedTo <http://aksw.org/MarkusAckermann.rdf> ] ;
nif:annotationUnit [ 
nif:taNerdCoreClassRef nerd:Location ;
nif:confidence "0.85"^^xsd:decimal ;
prov:wasAttributedTo <http://aksw.org/MartinBruemmer.rdf> ] ;
nif:annotationUnit [ 
itsrdf:taIdentRef <http://dbpedia.org/page/Fulton_County,_Georgia> ;
nif:confidence "0.72"^^xsd:decimal ;
prov:wasAttributedTo http://aksw.org/MartinBruemmer.rdf ] .
```
ITS and NERD

• Internationalizing Tag Set (ITS) 2.0

• Provides metadata for internationalization, translation and localization of XML, HTML and XML Schema.

https://www.w3.org/TR/its20/, https://github.com/w3c/itsrdf

• NERD is an ontology that models a set of taxonomies of NE types and provides mappings between them.

http://nerd.eurecom.fr/ontology

• Coupled with NIF, ITS and NERD have been used to model in RDF stand-off annotations resulting from NER, EL and concept extraction tools.

<#char=2797,2811>
a nif:Phrase ;
nif:anchorOf "European Union" ;
nif:beginIndex "2797"^^xsd:nonNegativeInteger ;
nif:endIndex "2811"^^xsd:nonNegativeInteger ;
its:taClassRef nerd:Organization ;
its:taIdentRef bn:00021127n ;
its:taIdentRef dbpedia:European_Union;
OLiA

- [http://nachhalt.sfb632.uni-potsdam.de/owl/](http://nachhalt.sfb632.uni-potsdam.de/owl/)
- Set of ontologies for annotation schemes used in NLP, linked together under a common reference model of linguistic concepts.
  - e.g. Penn Treebank
- Integrated into NIF

```
<#char=1478,1484> nif:oliaLink penn:NNS .
```
- **http://lemon-model.net/**
- Ontology for modelling lexicons information relative to ontology classes/properties..
- Used in DBpedia, BabelNet, PreMOn
- Extended by Ontolex W3C group (work in progress)

:tuberculosis lemon:canonicalForm [  
  lemon:writtenRep "tuberculosis"@en ] ;  
lemon:sense [  

:consumption lemon:canonicalForm [  
  lemon:writtenRep "consumption"@en ] ;  
lemon:sense [  
  lemon:reference <http://dbpedia.org/resource/Tuberculosis> ;  
  lemon:context :antiquated ] ;  
lemon:sense [  
PreMON

- https://premon.fbk.eu/
- Set of RDF datasets for predicate lexicons encoded and mapped according to an ontology for modelling semantic classes and roles
- Lexicons covered:
  - PropBank
  - NomBank
  - VerbNet
  - FrameNet
  - Semlink
- Uses Lemon, NIF.

```turtle
# VerbNet classs
pm:vn32-wipe_manner-10.4.1 a pmovn:VerbClass .
pm:vn32-wipe_manner-10.4.1 rdfs:label "wipe_manner-10.4.1" .

# subclass:
pm:vn32-wipe_manner-10.4.1-1 a pmovn:VerbClass .
pm:vn32-wipe_manner-10.4.1-1 skos:broader pm:vn32-wipe_manner-10.4.1

#x member:
pm:v-trim ontolex:evokes pm:vn32-wipe_manner-10.4.1 .
```
## Models Overview

<table>
<thead>
<tr>
<th>Ontology</th>
<th>Information modelled</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIF Core Ontology</td>
<td>Stand-off annotations</td>
<td>• Exchange of information between text analysis services.</td>
</tr>
<tr>
<td>EARMARK</td>
<td></td>
<td>• Production of LOD corpora</td>
</tr>
<tr>
<td>ITS + NIF + NERD</td>
<td>Annotation of NEs and concepts</td>
<td></td>
</tr>
<tr>
<td>PROV-O + NIF</td>
<td>Provenance and confidence</td>
<td></td>
</tr>
<tr>
<td>OLiA</td>
<td>NLP annotation tagsets</td>
<td>• Extend ontologies and LD with language-related information.</td>
</tr>
<tr>
<td>lemon</td>
<td>Non-predicative Lexicons</td>
<td>• Exchange of information between text analysis services.</td>
</tr>
<tr>
<td>PreMOn</td>
<td>Predicative lexicons</td>
<td>• Production of LOD corpora</td>
</tr>
</tbody>
</table>
Topics of this part

• Porting linguistic data to RDF
  – Models
    • NIF, EARMARK
    • PROV-O, ITS and NERD, OLiA
    • lemon
    • PreMOn

• Mapping linguistic structures to SW representations
  – Some relevant background
    • DUL DnS, c.DnS, LMM
  – Tools
    • LODifier
    • talnSRL
    • PIKES
    • FRED
Mapping linguistic structures to OWL

- Common translation practices:
  - NEs -> owl:NamedIndividual
  - Entity resolution/coreference -> owl:sameAs
  - Term -> owl:Class or owl:ObjectProperty / owl:DatatypeProperty
  - Sense Disambiguation -> owl:equivalentClass
  - Binary relation/semantic role -> owl:ObjectProperty or owl:DatatypeProperty
  - Taxonomy -> rdfs:SubClassOf

- But that’s just half of the story...
Leveraging linguistic and ontology semantics (I)

- **E.g. uniform reified modelling**
  - frame -> owl:NamedIndividual & Frame class
  - frame element -> objectProperty
    - connects frame and frame element filler instances

- **John is tall.**
  - Example uniform reified modelling

- **John gave Mary a new book.**
  - Example uniform reified modelling
Leveraging linguistic and ontology semantics (II)

- E.g. distinguish between frames’ ontological types
  - frame categories: events, artifacts, attributes, etc.
  - frame-category tailored element representation

- John is tall.

- John gave Mary a new book.

- BUT John is the leader of the group.
Leveraging linguistic and ontology semantics

• Challenges

  – gap between NLP structures and formal semantics adopted by SW ontologies

  – variety of requirements and application domains motivate different choices/translation practises from NLP to RDF/OWL representations
Some background

- **Description & Situation** (DnS) pattern in DOLCE+DnS Ultralite (DUL)
  - “relational context” based on a “descriptive context” (e.g. Diagnosis, Plan)
  - reified n-ary relation representation

- E.g. a DiagnosedSituation is a context of observed entities on the basis of a Diagnosis (Description)
- E.g. a SensorObservation is a context of a sensor, sensing method, observed feature and observation value, etc. satisfying a respective Observation (Description)
c.DnS

• Semiotic triangle realisation
  – informational context -> expression layer
  – circumstantial context -> reference layer
  – social context -> interpreter layer
  – all contexts -> meaning layer

• Application to FrameNet
  – Frames: meanings & descriptions
  – Frame Elements: meanings & concepts
  – Lexical Units: meanings & descriptions

• Also for WordNet, VerbNet

(Gangemi, 2009)
Linguistic Meta-Model (LMM)

- OWL-DL ontology for semiotic-cognitive representation of lexical knowledge
  - aligned with DOLCE Ultra Light (DnS)
  - multilingual

- LMM1
- LMM2
- LMM Alignments
Linguistic Meta-Model (LMM)

- LMM1 formalises the semiotic triangle
  - Reference: all entities belonging to the universe of discourse
    - dul:PhysicalArtifact, Imm2:IndividualReference, Imm2:MultipleReference, dul:Situation
  - Meaning:
    - dul:Description, dul:Collection, dul:Situation, dul:Concept
  - Expression: social objects produced by agents

(Picca et al., 2008)
Linguistic Meta-Model (LMM)

• LMM2
  – extends LMM1 capturing specific linguistic constructs and references

• LMM Alignments
  – LMM1, LMM2, DUL
  – WordNet, FrameNet, SKOS...
    • specialising dul:Concept (e.g. WNsynset, FE)
    • specialising dul:specializes (e.g. WNHypernym)
Outline

• Porting linguistic data to RDF
  – Models
    • NIF, EARMARK
    • PROV-O, ITS and NERD, OLiA
    • lemon
    • PreMOn

• Mapping linguistic structures to SW representations
  – Some relevant background
    • DUL DnS, c.DnS, LMM
  – Approaches
    • LODifier
    • talnSRL
    • PIKES
    • FRED
LODifier

• Generate Linked Data from unstructured text
  – RDF graphs linked to DBpedia and WordNet

• **Goal**: capture information from NL text as Linked Data

• Design principles
  – CCG & DRS
  – Linking to LOD cloud
LODifier

• NLP and KE tasks
  – Syntactic parsing: CCG
  – Semantic parsing and coreference resolution: Boxer
  – NER/EL: Wikifier
  – WSD: UKB

• NL to SW
  – Porting of DRSs to RDF
LODifier

• DRSs to RDF mapping
  – discourse referent -> blank node
  – unary condition -> RDF typing
  – binary condition -> RDF triple <s, p, o>
  – embedded DRSs -> RDF reification

• Vocabularies
  – closed: Boxer (event, agent, named, etc.)
  – open: WordNet senses, DBpedia URIs
The New York Times reported that John McCarthy died. He invented the programming language LISP.
talnSRL

- Frame-based relation extraction from NL sentences
  - RDF graphs generates Linked Data graphs

- **Goal**: capture information from NL text as Linked Data, including NLP information

- Design principles
  - Dependency parsing
  - Linguistic frames
  - Linking to LOD cloud
  - Anchor textual annotations

- [http://dparse.multisensor.taln.upf.edu/demoMS/relExtr/](http://dparse.multisensor.taln.upf.edu/demoMS/relExtr/)
- [https://github.com/talnsoftware/FrameSemantics_Parser](https://github.com/talnsoftware/FrameSemantics_Parser)
talnSRL

• NLP and KE tasks
  – Dependency parsing & Semantic parsing: Mate Tools and SemLink mappings
  – NER, EL & WSD: Babelfy

• Vocabularies
  – OLIA
  – ITS, NERD, SKOS
  – FrameNet
  – BabelNet
  – NIF

• NL to SW
  – Porting to RDF
talnSRL

**Representation model**
- based on FrameNet v1.5 RDF (Nuzzolese, Gangemi and Presutti 2011)
- nif:oliaLinks extensions
  - leverage NIF and OFN
  - enrich with linguistic information
PIKES

- Frame-based KE for English texts; two phases
  - linguistic features extraction
  - knowledge distillation

- **Goal**: Decouple NLP from knowledge distillation

- Design principles
  - Dependency parsing
  - FrameNet frames
  - Three-layer representation
    - Anchor textual annotations

PIKES

• NLP and KE tasks:
  – Dependency parsing & SRL: Mate Tools and Predicate Matrix mappings
  – NER and coreference resolution: Stanford CoreNLP
  – EL: DBpedia Spotlight
  – WSD: UKB

• Vocabularies
  – PropBank, NomBank, VerbNet, FrameNet
  – SEM, SUMO, FOAF, KnowledgeStore
  – DBpedia, YAGO
  – NIF

• NL to SW
  – Re-engineering using SPARQL-like mapping rules
  – Post-processing for materialising and compacting the resulting KB
• Three layer representation model

• Inter-layer relations
  – denotes
  – implies
  – expresses
The New York Times reported that John McCarthy died. He invented the programming language LISP.
FRED

- Machine reader for the SW
  - produces RDF/OWL ontologies and Linked Data from natural language sentences

- **Goal:** maximise modelling choices wrt Semantic Web practices

- Design principles
  - CCG & DRS
  - FrameNet frames
  - Ontology design patterns
  - Linking to LOD cloud
  - Anchor textual annotations

- [http://wit.istc.cnr.it/stlab-tools/fred](http://wit.istc.cnr.it/stlab-tools/fred)
**FRED**

- **NLP and KE tasks**
  - Syntactic parsing: CCG
  - Semantic parsing: Boxer + SemLink
  - Coreference resolution: CoreNLP
  - NER & EL: Stanbol, TagMe
  - WSD: UKB

- **Vocabularies**
  - LMM, DUL
  - VerbNet and FrameNet
  - DBpedia, WordNetRDF, OntoWordNet
  - NIF, EARMAR for annotation anchoring
  - POS, quantifiers, Boxer & FRED built-ins

- **NL to SW**
  - re-engineering & refactoring (Semion)
FRED

• N-ary logical patterns
  – events (role labelling) -> dul:Event | boxing -> dul:Situation

• Compositional semantics, taxonomy induction
  – “The programming language LISP...” -> ProgrammingLanguage(LISP) & subclassOf(ProgrammingLanguage,Language)
  – “John, the mayor of the city, ...” -> mayorOf(John, city)

• RDF-oriented representation for modality and negation
  – “John doesn’t like apples” -> hasTruthValue(like1, false)
The New York Times reported that John McCarthy died. He invented the programming language LISP.
## Tools Overview

<table>
<thead>
<tr>
<th></th>
<th>LODifier</th>
<th>talnSRL</th>
<th>PIKES</th>
<th>FRED</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Modelling</strong></td>
<td>porting to RDF</td>
<td>porting to RDF</td>
<td>re-engineering</td>
<td>re-engineering, maximises</td>
</tr>
<tr>
<td><strong>Textual annotations grounding</strong></td>
<td>-</td>
<td>NIF</td>
<td>NIF</td>
<td>NIF, EARMARK</td>
</tr>
<tr>
<td><strong>Semantic parsing</strong></td>
<td>DRS with Boxer</td>
<td>Dependencies with MTT DSynt + SemLink mappings</td>
<td>Dependencies with Mate tolos SLR + Predicate Matrix mappings</td>
<td>DRS with Boxer + SemLink mappings</td>
</tr>
<tr>
<td><strong>Vocabularies</strong></td>
<td>• WordNet, DBpedia</td>
<td>• OLiA, ITS, NERD, SKOS, OFN, BabelNet</td>
<td>• SUMO, YAGO2, SEM, DCMI, FOAF, PreMOn, DBpedia</td>
<td>• DUL, LMM, OFN, DBpedia, WordNet, OntoWN</td>
</tr>
<tr>
<td><strong>Coreference</strong></td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Multi-linguality</strong></td>
<td>only English</td>
<td>only English</td>
<td>only English</td>
<td>only English</td>
</tr>
</tbody>
</table>
Summary

• Two perspectives
  – porting linguistic resources and annotations
    • interlinking, interoperability, sharing & reuse
  – capturing distilled knowledge
    • automated reasoning (OL&P, QA, semantic search/retrieval, etc.)

• Translating linguistic representations to ontological ones involves non-trivial choices
  – linguistic vs knowledge engineering considerations
  – dependencies on application context
PART IV – Applications & Evaluation
Outline

• Applications
  – Question Answering over SW data
  – Abstractive summarization

• Evaluation
  – Intrinsic vs Extrinsic
  – KE tools comparison (Gangemi 2013)
  – Individual Tool Evaluation
    • LODifier, FRED, PIKES, SMATCH
Applications

• Capturing information from NL texts as structured representations is relevant to a plethora of applications
  – Ontology population & learning
  – Semantic search & retrieval
  – Linked Data publishing
  – Opinion and sentiment analysis
  – **Question Answering**
  – Natural Language generation-oriented tasks
    • ontology (axioms) verbalisation
    • delivery of RDF/OWL content
    • abstractive summarisation of RDF/OWL content
  – ....
Question Answering (I)

• Structured knowledge made available in RDF/OWL KBs keeps growing fast!!!
  – LOD cloud > 30 billion RDF triples
  – public knowledge bases (e.g. DBpedia, Freebase)
  – proprietary KBs (clinical, daily activities, smart homes, etc.)

• NL interfaces for Question Answering
  • intuitive paradigms for accessing & querying
  • hide the complexity of formal representation/query languages
Question Answering (II)

• **Goal**: translate the NL user questions into structured queries and interpret them against the underlying KB

• **Challenge**: bridge the gap between the way users communicate with the system and the way domain knowledge is captured
  – formalise and interpret user questions
    • conceptual granularity mismatches
    • meaning variations
Question Answering (III)

- Most systems focus primarily on simple factoid questions
  - “Who is the daughter of Robert Kennedy married to?”
  - “Who created Goofy?”

and (more recently) questions including superlatives, quantification, etc.
  - “What is the second highest mountain on Earth?”
  - “Give me all cities in Germany.”
  - “Which countries have more than two official languages?”

- I.e. questions with binary relations & “light” linguistic constructions
  - the more complex the linguistic constructions, the more challenging the question analysis (syntactic attachment & scope ambiguities, etc.)

- Example systems
  - triple-based serialisation: AquaLog, PowerAqua, NLP-reduce, FREyA, etc.
  - compositional semantics: Pythia, etc.
Question Answering (IV)

• Little support for QA over conceptually rich KBs capturing n-ary relational contexts
  – e.g. DOLCE + DnS ontology design patterns

• Example domains
  – event modelling
    e.g. Event-Model-F patterns for participation, mereology, causality, correlation, etc.
  – context-aware activity/behaviour modelling
Question Answering - Evaluation

• Question Answering over Linked Data (QALD)
  – [http://qald.sebastianwalter.org](http://qald.sebastianwalter.org)
  – set of questions along with corresponding answers and manually specified SPARQL queries
  – evaluation metrics: precision & recall
Abstractive summarisation

• Extractive summarization
  1. Splits source texts into fragments, e.g. sentences.
  2. Evaluates relevance of each.
  3. Produces summary by selecting most relevant fragments.

• Abstractive summarization
  1. Analyses source texts using NLP and IE methods.
  2. Creates an intermediate representation of contents.
  3. Uses natural language generation methods to produce a summary
Abstractive summarization

• Intermediate representations have different degrees of abstraction:
  – Abstract Meaning Representation, e.g. Liu et al. 2015
  – Discourse trees, e.g. Gerani et al. 2014
  – Dependency trees, e.g. Cheung and Penn 2014
  – Word co-occurrence graphs, e.g. Ganesan et al. 2010

• Most abstractive summarizers use linguistic semantic structures
  – Yet ontological representations open the door to reasoning!
    • E.g. Temporal reasoning can be used to order events in the summary.
    • E.g. Reasoning about models of user profiles and preferences can be used to create tailored summaries.
Abstractive summarization with AMR structures (Liu et al. 2015)

- **Resources:**
  - Statistical AMR parser JAMR trained on AMR Bank.
  - Dataset of documents, summaries and gold-standard AMR annotations of both (AMR Bank).

- AMR sentence graphs are merged into a single document graph by collapsing common NE nodes.

- Summary is a subgraph of the document graph.

- Summarization addressed through features on the graph with scores estimated from the dataset using ILP.

Sentence A: I saw Joe’s dog, which was running in the garden.
Sentence B: The dog was chasing a cat.
Outline

• Applications
  – Question Answering over SW data
  – Abstractive summarization

• Evaluation
  – Intrinsic vs Extrinsic
  – KE tools comparison (Gangemi 2013)
  – Individual Tool Evaluation
    • LODifier, FRED, PIKES, SMATCH
Evaluation

- How to evaluate a relation extraction system?

- Intrinsic vs extrinsic evaluation
  - Extrinsic evaluation
    - performance of downstream application as indicator of quality
      - e.g. question answering, ontology population, summarization.
  - Intrinsic evaluation
    - compare extracted relations to a manually annotated gold standard in terms of precision and recall
KE tools comparison (Gangemi 2013)

- Unified framework for comparing multiple KE tools
- A reference knowledge space is created by converting the results of each tool to RDF/OWL expressions and merging the results.
- Evaluation metrics: precision, recall, f-score and accuracy
Comparison of KE tools (Gangemi 2013)

- Comparison of n-ary relations as overlap of unlabeled binary relations.

"it plans to blacklist the Nusra Front as a terrorist organization"

- (ReVerb): no extraction
- (Alchemy): plans to blacklist(it, the Nusra Front as a terrorist organization)
- (CiceroLite): plans to blacklist(it, front, (AS) a terrorist organization)
- (FRED): experiencer(plan_3, United_States) ; theme(plan_3, blacklist_3);
  agent(blacklist_3, United_States) ; patient(blacklist_3, NusraFront) ;
  as(blacklist_3, organization_3) ; TerroristOrganization(organization_3)

<table>
<thead>
<tr>
<th>RelEx Tool</th>
<th>p</th>
<th>r</th>
<th>F1</th>
<th>a</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alchemy</td>
<td>.69</td>
<td>.25</td>
<td>.37</td>
<td>.30</td>
</tr>
<tr>
<td>CiceroLite</td>
<td>.90</td>
<td>.20</td>
<td>.33</td>
<td>.25</td>
</tr>
<tr>
<td>FRED</td>
<td>.84</td>
<td>.82</td>
<td>.83</td>
<td>.82</td>
</tr>
<tr>
<td>ReVerb</td>
<td>.67</td>
<td>.23</td>
<td>.34</td>
<td>.27</td>
</tr>
</tbody>
</table>

Table 9: Comparison of relation extraction tools.

- “There are important cohesion aspects here that are hardly caught by means of simple triple patterns”
Evaluation of LODifier

- Extrinsic evaluation: document similarity task
  - subset of 183 document pairs belonging to same topic, taken from TDT-2 benchmark dataset
  - baseline: set of similarity measures between documents that ignore relations: random, bag-of-words, bag-of-URI (NEs and WordNet)
Evaluation of LODifier

- Similarity measures based on similarity between graphs extracted by LODifier for each document
  - graph isomorphism too computationally expensive
  - approximated similarity using overlap in set of shortest paths between relevant entities

<table>
<thead>
<tr>
<th>Model</th>
<th>Similarity measures without structural knowledge</th>
<th>Similarity measures with structural knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Model</td>
<td>proSim_len (k=6, Variant 1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>proSim_len (k=6,  Variant 3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>proSim_len (k=8,  Variant 2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>proSim_aplen (k=6, Variant 3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>proSim_aplen (k=8, Variant 3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>proSim_aplen (k=10, Variant 3)</td>
</tr>
<tr>
<td></td>
<td>Similarity measures without structural knowledge</td>
<td>79.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>81.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>77.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>79.2</td>
</tr>
<tr>
<td></td>
<td>proSim_len (k=6,  Variant 3)</td>
<td>81.5</td>
</tr>
<tr>
<td></td>
<td>proSim_len (k=8,  Variant 3)</td>
<td>80.3</td>
</tr>
<tr>
<td></td>
<td>proSim_len (k=10, Variant 3)</td>
<td>80.0</td>
</tr>
<tr>
<td></td>
<td>proSim_aplen (k=6, Variant 3)</td>
<td>80.4</td>
</tr>
<tr>
<td></td>
<td>proSim_aplen (k=8, Variant 3)</td>
<td>81.1</td>
</tr>
<tr>
<td></td>
<td>proSim_aplen (k=10, Variant 3)</td>
<td>80.5</td>
</tr>
</tbody>
</table>
Evaluation of FRED

• Intrinsic evaluation
  – gold standard of 1214 sentences randomly selected from FrameNet annotated corpus; one frame per sentence; only verbs
  – given a sentence, when a predicted frame matches gold-standard full score (1), when semantically close partial score (0.8)
  – metrics: precision & recall of extracted frames
    • comparison against *Semafor*

<table>
<thead>
<tr>
<th>Tool</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boxer</td>
<td>75.320</td>
<td>57.519</td>
<td>65.227</td>
</tr>
<tr>
<td>Semafor</td>
<td>75.325</td>
<td>74.797</td>
<td>75.060</td>
</tr>
</tbody>
</table>
Evaluation of PIKES

• Intrinsic evaluation
  – gold standard of 8 sentences manually annotated by two annotators
  – metrics: precision and recall in extracted entities, frames attributes, and their relations

  – Gold and system outputs seen as a graph G where:
    • nodes: entities, frames, attributes annotated and matched against entries in DBpedia and types in VerbNet, FrameNet and PropBank
    • edges: coreference relations, instance-attribute associations, and frame-argument participation with VN/FN/PB/NB thematic roles
Evaluation of PIKES against gold

- Evaluations for each component of G and G_S:
  - Nodes/Instances
  - (unlabeled) Edges
  - (labeled) Triples

- Triples are considered divided by category:
  - Links to DBpedia, VN/FN/PB/NB classes, VN/FN/PB/NB participation relations, coreference relations

- True positives, false positives, true negatives and false negatives over sets of components

```
<table>
<thead>
<tr>
<th>Component</th>
<th>N_G</th>
<th>P</th>
<th>R</th>
<th>F_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instances</td>
<td>153</td>
<td>.943</td>
<td>.967</td>
<td>.955</td>
</tr>
<tr>
<td>Triples</td>
<td>596</td>
<td>.713</td>
<td>.508</td>
<td>.594</td>
</tr>
<tr>
<td>DBpedia links</td>
<td>18</td>
<td>.700</td>
<td>.778</td>
<td>.737</td>
</tr>
<tr>
<td>types (VN)</td>
<td>44</td>
<td>.706</td>
<td>.545</td>
<td>.615</td>
</tr>
<tr>
<td>types (FN)</td>
<td>53</td>
<td>.731</td>
<td>.358</td>
<td>.481</td>
</tr>
<tr>
<td>types (PB)</td>
<td>55</td>
<td>.844</td>
<td>.717</td>
<td>.776</td>
</tr>
<tr>
<td>types (NB)</td>
<td>37</td>
<td>.690</td>
<td>.784</td>
<td>.734</td>
</tr>
<tr>
<td>roles (VN)</td>
<td>94</td>
<td>.742</td>
<td>.489</td>
<td>.590</td>
</tr>
<tr>
<td>roles (FN)</td>
<td>108</td>
<td>.500</td>
<td>.259</td>
<td>.341</td>
</tr>
<tr>
<td>roles (PB)</td>
<td>119</td>
<td>.829</td>
<td>.571</td>
<td>.677</td>
</tr>
<tr>
<td>roles (NB)</td>
<td>55</td>
<td>.627</td>
<td>.582</td>
<td>.604</td>
</tr>
<tr>
<td>owl:sameAs</td>
<td>15</td>
<td>.714</td>
<td>.333</td>
<td>.455</td>
</tr>
<tr>
<td>Edges</td>
<td>171</td>
<td>.891</td>
<td>.766</td>
<td>.824</td>
</tr>
</tbody>
</table>
```
Evaluation of PIKES against FRED

• A modified gold graph $G'$ is produced without the information that FRED doesn’t produce:
  – PropBank and NomBank types and roles
  – FrameNet roles
  – Nominal predicates

• An additional graph $G''$ is produced by merging the results of FRED and PIKES

<table>
<thead>
<tr>
<th>Component</th>
<th>$N_{G'}$</th>
<th>FRED vs $G'$</th>
<th>PIKES vs $G'$</th>
<th>$N_{G''}$</th>
<th>FRED vs $G''</th>
<th>PIKES vs $G''$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F₁</td>
<td>P</td>
<td>R</td>
<td>F₁</td>
</tr>
<tr>
<td>Instances</td>
<td>137</td>
<td>.930</td>
<td>.869</td>
<td>.898</td>
<td>.937</td>
<td>.978</td>
</tr>
<tr>
<td>Triples</td>
<td>166</td>
<td>.543</td>
<td>.416</td>
<td>.471</td>
<td>.713</td>
<td>.554</td>
</tr>
<tr>
<td>DBpedia links</td>
<td>18</td>
<td>.615</td>
<td>.444</td>
<td>.516</td>
<td>.700</td>
<td>.778</td>
</tr>
<tr>
<td>types (VN)</td>
<td>31</td>
<td>.593</td>
<td>.516</td>
<td>.552</td>
<td>.667</td>
<td>.581</td>
</tr>
<tr>
<td>types (FN)</td>
<td>26</td>
<td>.550</td>
<td>.423</td>
<td>.478</td>
<td>.762</td>
<td>.615</td>
</tr>
<tr>
<td>roles (VN)</td>
<td>76</td>
<td>.547</td>
<td>.382</td>
<td>.450</td>
<td>.722</td>
<td>.513</td>
</tr>
<tr>
<td>owl:sameAs</td>
<td>15</td>
<td>.357</td>
<td>.333</td>
<td>.345</td>
<td>.714</td>
<td>.333</td>
</tr>
<tr>
<td>Edges</td>
<td>155</td>
<td>.869</td>
<td>.555</td>
<td>.677</td>
<td>.937</td>
<td>.768</td>
</tr>
</tbody>
</table>

(c)
SMATCH

- AMR evaluation metric as degree of overlap between two whole-sentence semantic structures
- Measures precision, recall, and f-score of the triples in the second AMR against the triples in the first AMR, i.e., the amount of propositional overlap
- Conjunction of logical propositions, or triples:
  - Maximum f-score obtainable via a one-to-one matching of variables between the two AMRs

```
 instance(a, want-01) ∧
 instance(b, boy) ∧
 instance(c, go-01) ∧
 ARG0(a, b) ∧
 ARG1(a, c) ∧
 ARG0(c, b)
```

```
 x=a, y=b, z=c: 4 4/5 4/6 0.73
 x=a, y=c, z=b: 1 1/5 1/6 0.18
 x=b, y=a, z=c: 0 0/5 0/6 0.00
 x=b, y=c, z=a: 0 0/5 0/6 0.00
 x=c, y=a, z=b: 0 0/5 0/6 0.00
 x=c, y=b, z=a: 2 2/5 2/6 0.36
```

smatch score: 0.73

“the boy wants to go”    “the boy wants the football”
Conclusions

• NLP and SW are starting to get along better
  – yet, still a long way to go.....

• Challenges
  – Performance of NLP tools
    • error propagation
  – No principled way of converting NL structures to OWL
    • no straightforward mappings between different conversions
  – Evaluation is not straightforward
QUESTIONS?

This tutorial is supported by the European commission under the contract numbers FP7-ICT-610411 and H2020-645012-RIA
References (I)


References (II)


References (III)


References (IV)


References (V)