Extracting Entity-Relationship Triples from Text

Liane Guillou

7/6/2018

What is Natural Language Processing (NLP)?

- Area of computer science / artificial intelligence
- Focus: human-computer interaction using natural language e.g. English, German, Chinese, Russian, etc.
- Communication modes: text or speech
- Many applications:
  - Machine Translation (Google Translate)
  - Question Answering (IBM's Watson)
  - Sentiment Analysis (e.g. for marketing)
  - Speech Recognition (in Siri)
  - Speech Synthesis (Alexa)
  - etc.

NLP at Edinburgh University

Part of Institute for Language, Cognition and Computation (ILCC):

- 39 Academic staff / senior researchers
- 30 Researchers / postdocs
- 75 PhD students

Courses:

- Masters with a specialism in NLP
- Undergraduate courses from year 2 onwards

Python widely used

SEMANTAX Project

- 5 year research project
- 1 professor, 3 postdocs, 4 PhD students
- Focus:
  - Develop and apply a form-independent semantics
  - Encode information in a natural-language compatible way
- Areas of interest:
  - Knowledge graph construction
  - Question answering
  - Parsing
  - Multilingual / cross-lingual aspects

Entity-Relationship Triple Extraction: Overview

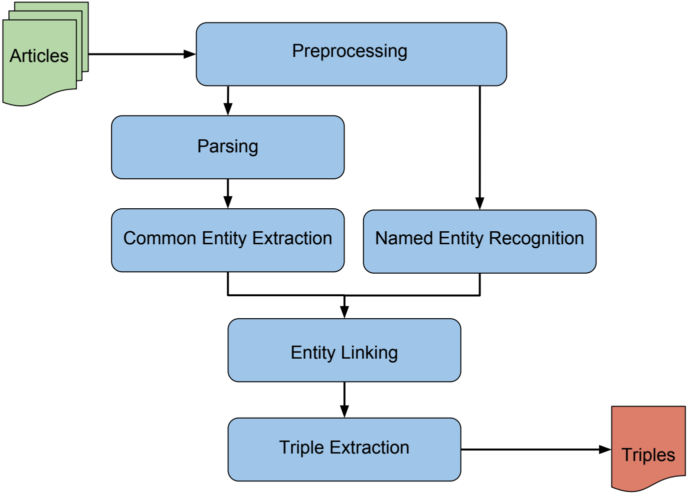
- Also known as “Binary Relations”
- Two entities, with a relationship that holds between them
- Format: {entity1, relationship, entity2}

Example Triple

Text:	Edinburgh is located in Scotland.
Triple:	{Edinburgh, is located in, Scotland}

- Triples extracted from text express *facts*
- Existing English pipeline: triple extraction + graph construction
- Focus of this talk: German pipeline

Pipeline Architecture



Pipeline

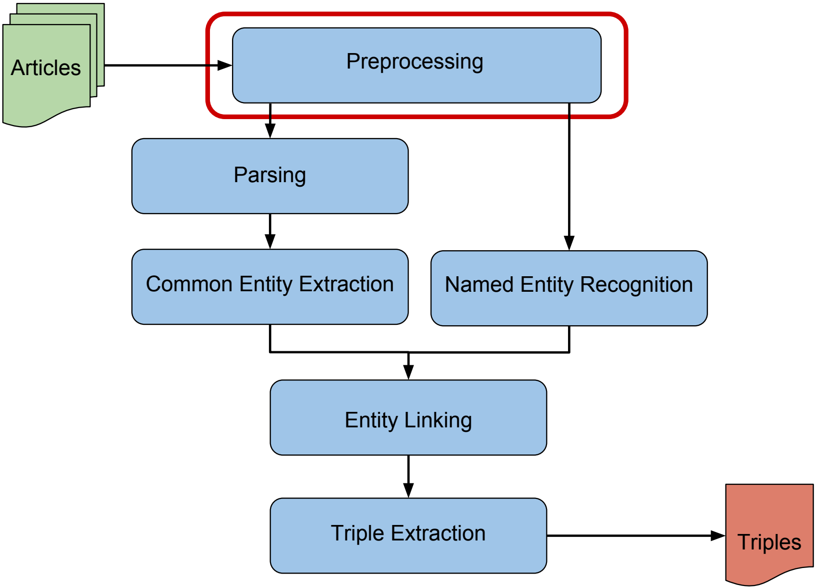
- Designed to extract all possible entity-relationship triples
  In contrast with machine-learning methods: typically focus on a small set of patterns
- Combines external tools + linguistic rules
- Downstream applications:
  - **Machine Translation evaluation:** does the German translation capture the meaning of the original English text? (compare triples)
  - **Cross-lingual question answering:** ask a question in German about an English news article (construct knowledge graph from triples)

Example Sentence

(1) Angela Merkel wuchs in der DDR auf
Angela Merkel grew in the DDR up
'Angela Merkel grew up in the DDR'

Particle verbs

aufwachsen = to grow up (infinitive)
wuchs auf = grew up (past tense)
Also exist in English e.g. "put up"


Articles
Preprocessing
Parsing
Common Entity Extraction
Named Entity Recognition
Entity Linking
Triple Extraction
Triples

Sentence Segmentation

- Split documents / paragraphs into sentences
- Why? Simpler to work at the sentence level

Paragraph/Document

Angela Dorothea Merkel ist eine deutsche Politikerin (CDU). Am 14. März 2018 wurde Merkel vom Bundestag zum vierten Mal zur Bundeskanzlerin gewählt. Angela Merkel wuchs in der DDR auf...

Sentences

(1) Angela Dorothea Merkel ist eine deutsche Politikerin (CDU).
(2) Am 14. März 2018 wurde Merkel vom Bundestag zum vierten Mal zur Bundeskanzlerin gewählt.
(3) Angela Merkel wuchs in der DDR auf...

- Using NLTK's PunktTokenizer model

Word Tokenisation

- Split sentences into words / tokens
- Why? Punctuation is not part of a word

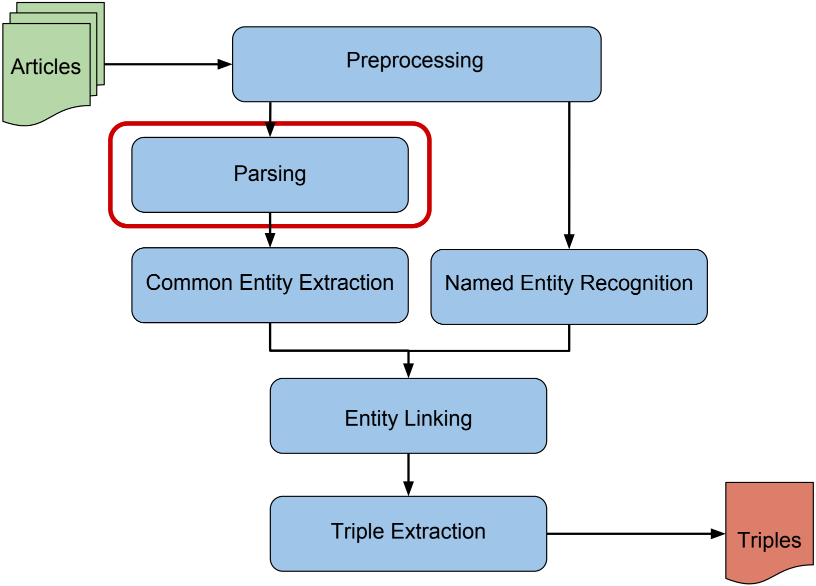
Sentence

Merkel wuchs in der DDR auf.

Tokens

Merkel
wuchs
in
der
DDR
auf
.

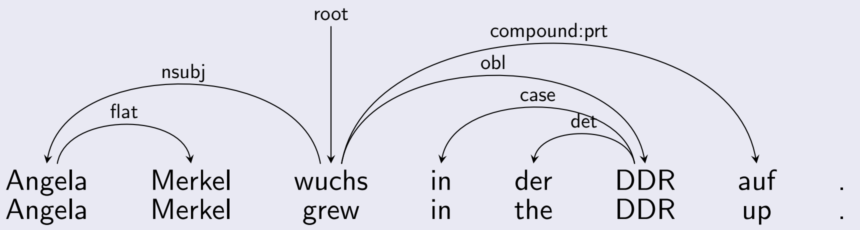
- Using Python module: UDPipe


Articles
Preprocessing
Parsing
Common Entity Extraction
Named Entity Recognition
Entity Linking
Triple Extraction
Triples

Dependency Parsing

- Sentence represented as a tree
- Tree has a root (typically the *main* verb)
- Relation between two words: *labelled* arc from *head* to *dependent*
- Provides the relationships for triples

Dependency Tree



Universal Dependencies

- Framework for cross-linguistically consistent grammatical annotation
- Key idea: set of core dependencies, universal to all languages (e.g. *nsubj*)
- Some languages may require extra dependencies:
  200 extra language-specific dependencies
- Treebanks for 73 languages
- Useful for cross/multi-lingual work:
  build pipelines for other languages and use similar parser + rules to extract relations

`http://universaldependencies.org/`

Unstable Parser

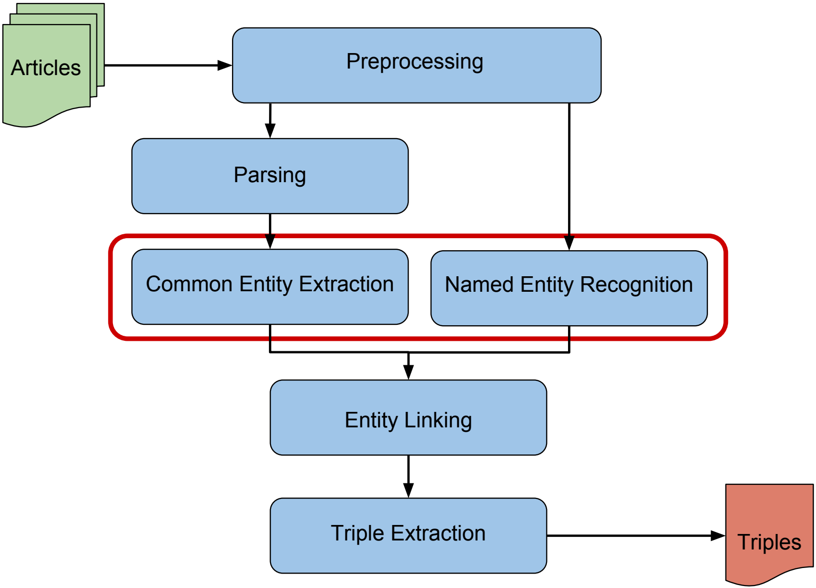
- Neural Network based dependency parser (in Python)
- Best system: CoNLL 2017 shared task on universal dependency parsing
- Can download any Universal Dependency treebank and train a parser
- German parser can be trained overnight
- Input: tokenised text (from UDPipe module)

Unstable Parser

Parser Output: CoNLL Format

			POS tag			
ID	Word	Lemma	Coarse	Fine	Head	Dependency-rel.
1	Angela	Angela	PROPN	NE	3	nsubj
2	Merkel	Merkel	NOUN	NN	1	flat
3	wuchs	wachsen	VERB	VVFIN	0	root
4	in	in	ADP	APPR	6	case
5	der	der	DET	ART	6	det
6	DDR	DDR	PROPN	NE	3	obl
7	auf	auf	ADP	PTKVZ	3	compound:prt
8	.	.	PUNCT	$.	3	punct

https://github.com/tdozat/Parser-v2


Articles
Preprocessing
Parsing
Common Entity Extraction
Named Entity Recognition
Entity Linking
Triple Extraction
Triples

Named Entity Recognition

- Provides the entities for the triples
- **Named Entity**: a real-world object that can be denoted with a proper name
  e.g. a person, location, organisation, product, etc.
- **Named Entity Recognition**: finding Named Entities in text

Stanford NER Output

[Angela Merkel]	wuchs	in	der	[DDR]	auf	.	
PERSON				LOCATION			

- Using Python module: sner (wrapper for Stanford NER)

Common Entity Extraction

- Provides the entities for the triples
- **Common / General Entity**: a common real-world *thing*
- Extracted using Parser output + Part-of-Speech Tags
  - Find spans of *nouns*

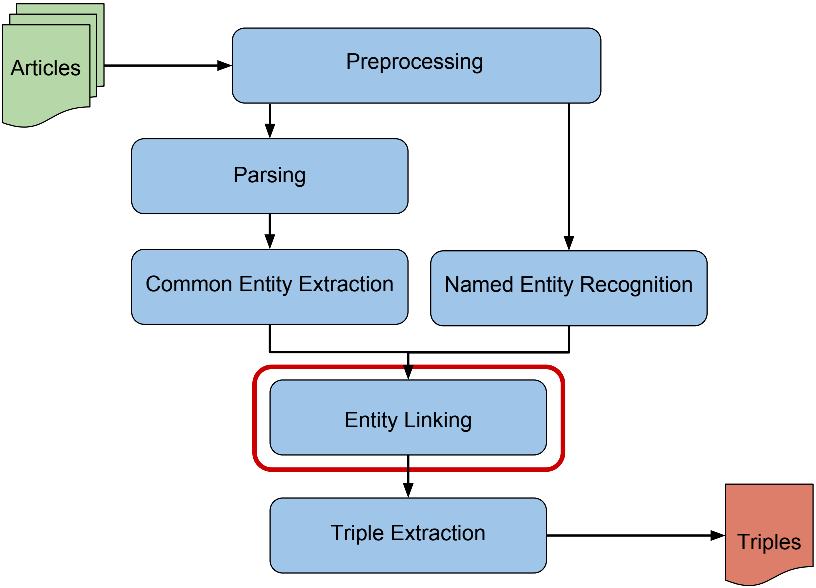
A New Example

(2)

Eine	Katastrophe	für	die	Parteichefin
DET	NOUN	ADP	DET	NOUN
A	disaster	for	the	party leader

'A disaster for the party leader'

Common entities: Eine [Katastrophe] für die [Parteichefin]


Articles
Preprocessing
Parsing
Common Entity Extraction
Named Entity Recognition
Entity Linking
Triple Extraction
Triples

Entity Linking

Map “Angela Dorothea Merkel”, “Angela Merkel”, “Merkel” to same entity

Input

'⟨entity⟩Angela Merkel⟨/entity⟩ wuchs in der ⟨entity⟩DDR⟨/entity⟩ auf .'

Output


```
{
  "disambiguatedURL": "http://de.dbpedia.org/resource/Angela_Merkel",
  "offset": 13,
  "namedEntity": "Angela Merkel",
  "start": 1
}

{
  "disambiguatedURL": "http://de.dbpedia.org/resource/Deutsche_Demokratische_Republik",
  "offset": 3,
  "namedEntity": "DDR",
  "start": 28
}
```


Using AGDISTIS: `http://aksw.org/Projects/AGDISTIS`

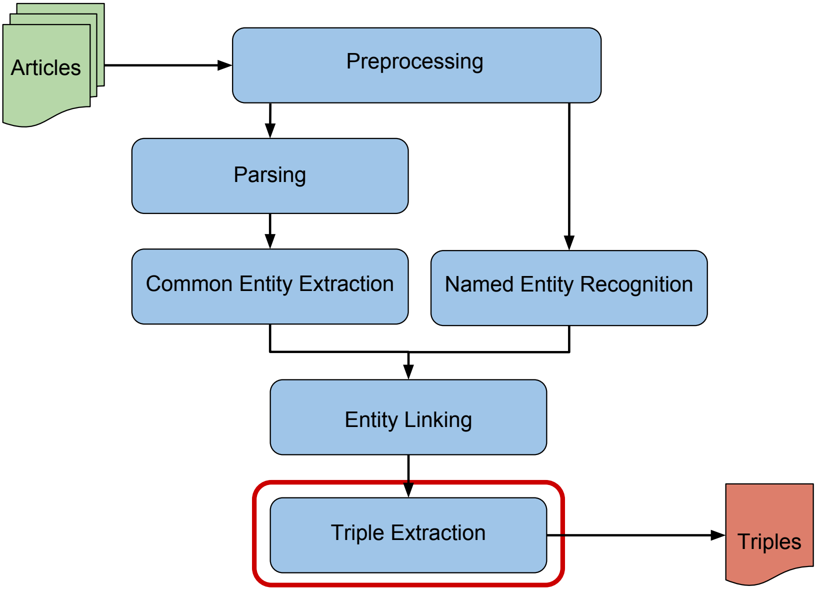
Entity Types

- We also wish to know the *semantic type* of each entity
  to encode information such as: a PERSON may visit a LOCATION
  to build knowledge graphs for downstream applications e.g. QA
  - Named Entity Recogniser gives us basic types: PERSON, LOCATION, ORGANISATION, MISC.
  - Entity Linker gives us CHEMICAL, LIVING_THING, etc.
- Types help to align English and German triples for cross-lingual QA
- Map DBPedia URL → first level of FIGER type system

FIGER Types

PERSON/politician	Angela_Merkel
LOCATION/country	Deutsche_Demokratische_Republik
BUILDING/airport	Hellinikon_Airport

https://github.com/xiaoling/figer


Articles
Preprocessing
Parsing
Common Entity Extraction
Named Entity Recognition
Entity Linking
Triple Extraction
Triples

Extracting Entity-Relation Triples

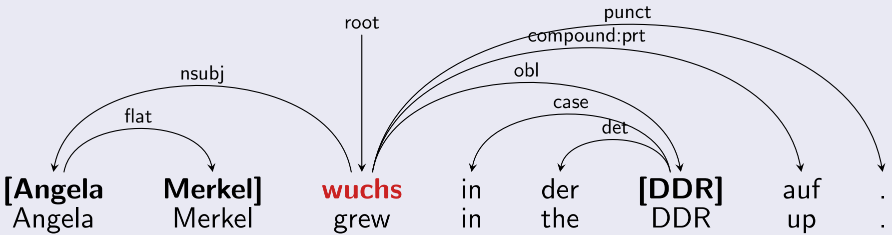
Combines: dependency parse + linked entities + rules

Basic steps: for each pair of entities

1. find entities in parse tree
2. if verb links the entities:
    1. find dependency between entity A and verb
    2. find dependency between entity B and verb
    3. if entity A in subject position and B in object position (or vice versa):
        1. find lemma of verb
        2. extract: subject-(verb_lemma)-object triple

Example Extraction

Dependency Tree



Entities

Angela Merkel, PERSON, http://de.dbpedia.org/resource/Angela_Merkel
DDR, LOCATION, http://de.dbpedia.org/resource/Deutsche_Demokratische_Republik

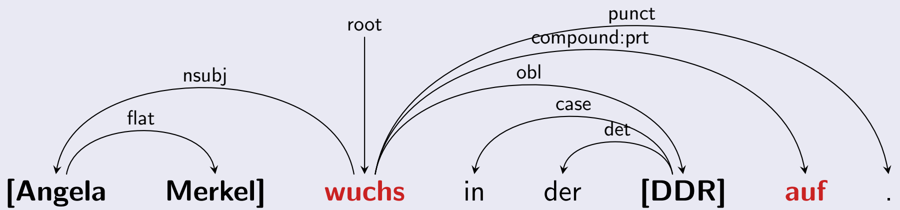
Triple

{Angela Merkel, wachsen, DDR} :: #PERSON:#LOCATION **(?)**

Particle Verbs

Reminder: aufwachsen (past tense: wuchs auf) is a particle verb

Dependency Tree



Triple

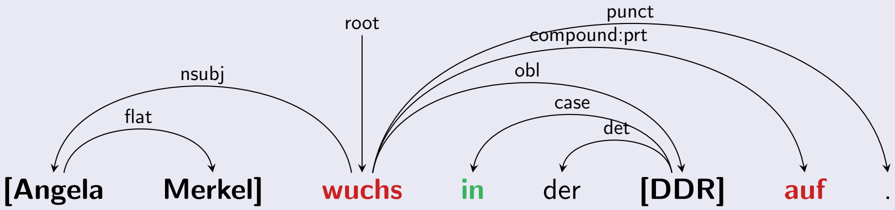
{Angela Merkel, wachsen auf, DDR} ✓

Prepositions

Preposition choice changes verb meaning

Verb	Preposition	Meaning
sich freuen	auf	to look forward to
sich freuen	über	to be pleased with

Dependency Tree



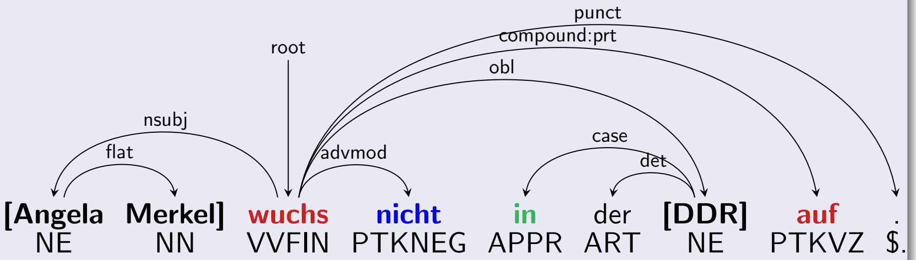
Triple

{Angela Merkel, wachsen auf in, DDR} ✓

Negation

What if: Angela Merkel *hadn't* grown up in the DDR

Dependency Tree



Triple

NEG_{Angela Merkel, wachsen auf in, DDR} ✓

Active-to-Passive Conversion

Passive Example

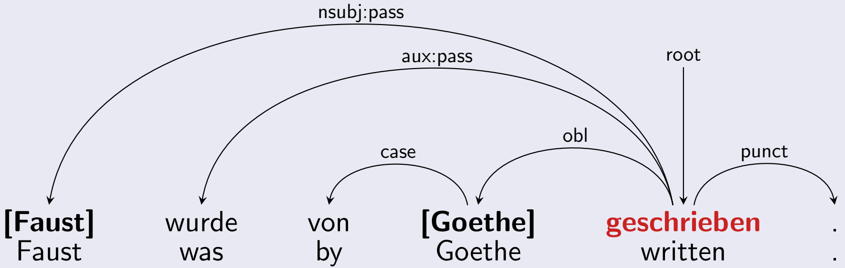
(3) Faust wurde von Goethe geschrieben
Faust was by Goethe written

'Faust was written by Goethe'

Active: Goethe wrote Faust (swap subject and object)

Active-to-Passive Conversion

Dependency Tree



Triple

{Goethe, schrieben, Faust} ✓

Some Sample Stats

- 10 news articles
- 362 sentences
- 105 entity-relationship triples
  - Many are good
  - Some are bad - thresholding will help

Summary

- Constructed triple extraction pipeline for German
  in line with existing English pipeline
- Next steps:
  - Refine linguistic rules for triple extraction / increase coverage
  - Optimisation: identify and resolve bottlenecks
  - Integrate with language-independent graph-generation pipeline
  - Align English and German triples / knowledge graphs (for cross-lingual work)
  - Evaluate triple extraction via downstream tasks
- Downstream applications:
  - Machine translation evaluation
  - Cross-lingual question answering


GOT A
QUESTION