Relation Extraction

COMP61332: Text Mining
Week 2
Riza Batista-Navarro

Relation Extraction

Discerning **relationships** amongst **entities** in piece of text

[American Airlines], a unit of [AMR Corp.], immediately matched the move, spokesman [Tim Wagner] said. [United], a unit of [UAL Corp.], said the increase took effect Thursday and applies to most routes.

Relationships:

Tim Wagner *spokesman for* American Airlines

United *unit of* UAL Corp.

American Airlines *unit of* AMR Corp.

Relationship types: Predefined

Examples

Affiliations

Personal: *married to, mother of*
Organisational: *spokesman for, president of*
Artifactual: *owns, invented, produces*

Geospatial

Proximity: *near*
Directional: *southeast of*

Part-of

Organisational: *unit of*

Approaches

Pattern-based

Statistical/machine learning-based

Pattern-based Approaches

Using **regular expressions** (regexes)

Example: extract *airline-hub cities* relations

> regex: */* has a hub at *\/*
>
> would match: *KLM has a hub at Amsterdam.*
>
> but also false positives: *The wheel has a hub at its centre.*

Pattern-based Approaches

Regex can be modified to put entity constraints

/[ORGANISATION] has a hub at [LOCATION]/

but still problematic as it would miss:

easyJet has established a hub at Liverpool.

Ryanair has a continental hub at Charleroi, Belgium.

Pattern-based Approaches

How can we reduce false negatives?

Two options:

- Relaxing the pattern (to skip certain words)
- Expand the set of patterns (while keeping precision) using bootstrapping

Bootstrapping

1. Start with a small set of **seed patterns** and **seed tuples**

   seed pattern: */* has a hub at *\/*

   seed tuple: *Ryanair - Charleroi*

2. Search for the terms in the seed tuple, within a corpus of documents

   Budget airline Ryanair which uses Charleroi as a hub, scrapped all weekend flights out of the airport.

   All flights in and out of Ryanair's Belgian hub at Charleroi airport were grounded.

   A spokesman at Charleroi, a main hub for Ryanair, estimated that 8000 passengers had already been affected.

Bootstrapping

3. Use the search results to extract new patterns

   Budget airline Ryanair which uses Charleroi as a hub, scrapped all weekend flights out of the airport.

   /[ORGANISATION], which uses [LOCATION] as a hub/

   All flights in and out of Ryanair's Belgian hub at Charleroi airport were grounded.

   /[ORGANISATION]'s hub at [LOCATION]/

   A spokesman at Charleroi, a main hub for Ryanair, estimated that 8000 passengers had already been affected.

   /[LOCATION], a main hub for [ORGANISATION]/

Bootstrapping

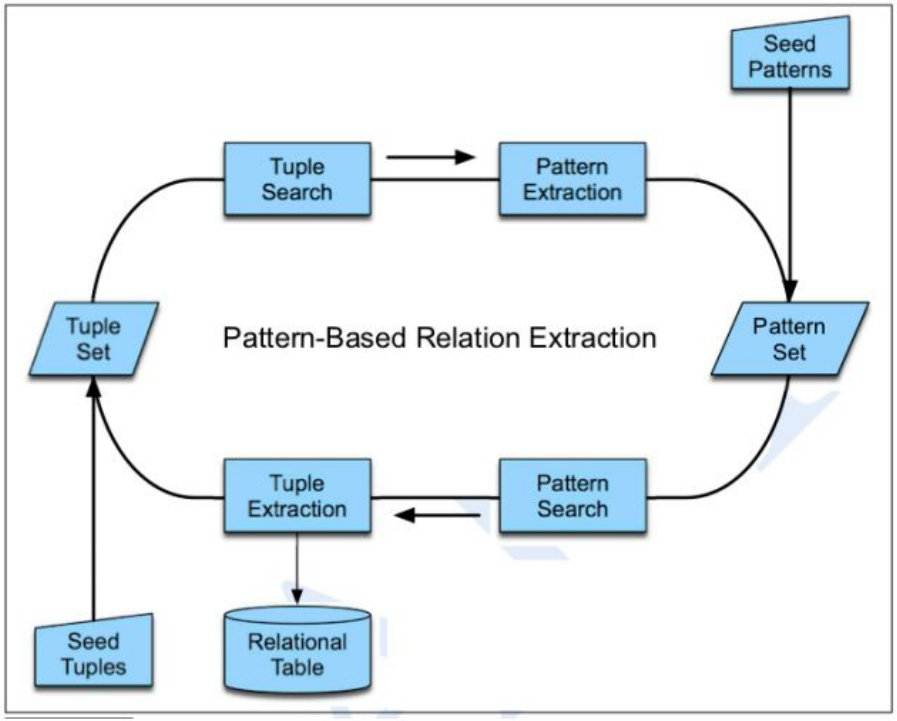
3. Use the search results to extract new patterns

   Considerations:

   - context before the first entity
   - context between the two entities
   - context after the second entity

Bootstrapping

4. Search for additional tuples using the new patterns (to grow set of tuples)



Machine learning-based Approaches

Requires manually labelled (annotated) text

Two subtasks:

- **Detecting** if a relationship is present between a given pair of entities
- **Classifying** the relationship

Machine learning-based Approaches

Subtask 1: Do two entities participate in a relation?

A binary classification task

Positive examples: entities that are annotated as being related

Negative examples: entities (within the same sentence) not annotated as being related

Machine learning-based Approaches

Subtask 2: What type of relation holds?

A multi-class classification task

Some methods incorporate/jointly learn Subtask 1 (including a *no_relation* class)

Feature types

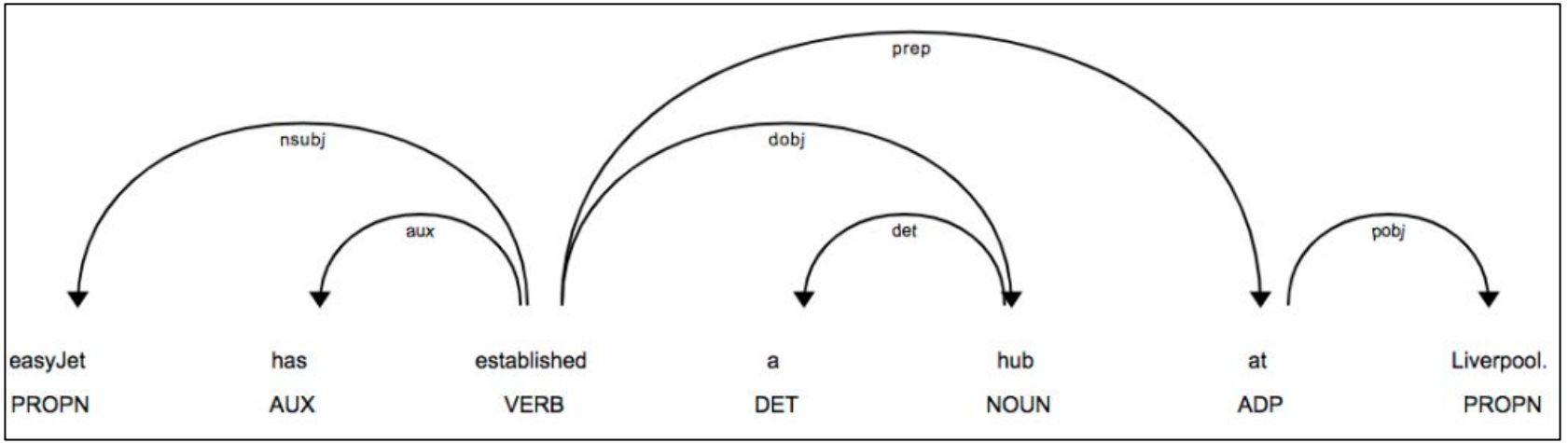
named entity features: entity type, head word

context: words between, words before the first entity, words after the second entity (within a fixed window)

syntactic structure: dependency paths

Dependency Paths

Useful in both machine learning-based and pattern-based relation extraction



For the *easyJet - has city hub at - Liverpool* relationship:

easyJet ← established → at → Liverpool