SNLP Mini Project

Group: semicolonMissing

Members: Kadiray Karakaya

Outline

- Initial Approach
- Problems
- Final Approach & Thoughts
- Task Distribution

Initial Approach

- Extract information from the input
- Find related wikipedia article of the input’s subject
- Extract information from the related wikipedia article
- Compare similarity for the extracted informations

Initial Approach

- Extract information from the input
  - Use Stanford CoreNLP Open Information Extraction annotator to extract relation as:
    - Subject
    - Object
    - Relation

Initial Approach

- Example Input:
  - Elizabeth Taylor's death place is London

- Extracted triples (Subject, Relation, Object):
  - Triple1: Elizabeth Taylor, have, death place
  - Triple2: Elizabeth Taylor’s deatplace, be, London
  - Hence: Elizabeth Taylor, DIE_IN, London

Initial Approach

- Use Subject to find Wikipedia article
  - The article’s url is mostly found by replacing the spaces with an underscore
  - ie. «Elizabeth Taylor» -> «Elizabeth_Taylor»
  - https://en.wikipedia.org/wiki/Elizabeth_Taylor

Initial Approach

- Again use Stanford Open IE to extract relations from the article.
- This generates a lot of candidate triples to compare.
- Reduce the candidate triples to have a better runtime for similarity comparison:
  - Find the Subjects from the article that contains the Subject from the input
  - Find the Objects from the article that contains the Object from the input

Initial Approach

- Run a similarity measure between good candidate triples against the input triples
- We used Carnegie Mellon's WordNet Similarity for Java library, it offers many methods like:
  - Wu-Palmer
  - Leacock-Chodorow
  - Lesk
  - etc

First Problems Arise

- Similarity for a correct sentence:
  - Input: "Elizabeth Taylor", "die in", " London"
  - Article: "Elizabeth Taylor", "die in", " London "
  - Wu-Palmer similarity: 1.0

First Problems Arise

- Similarity for an incorrect sentence:
  - Input: "Elizabeth Taylor", "die in", "Paris"
  - Article: "Elizabeth Taylor", "die in", "London"
  - Wu-Palmer similarity: 0.84

  - It is still a good score, eventhough it completely false!

More Problems

1. Not every Subject and Object can be found by Stanford Open IE
2. It is not always easy to conclude a final relation (by chaining all the triples)
3. Using wikipedia article body takes too much time to process by Stanford library

We need another approach

- Find patterns from the input to conclude a final relation
  - (solves 2. problem)
- Everything except the relation pattern leaves Subject and Object
  - (solves 1. problem)
- Parse wikipedia page to access more relevant (critical) information

Ideal Goal

- Train a Learning algorithm by labeling patterns with relations:
  - Count Basie's death place is Hollywood, Florida
  - Adam Smith's death place is Buffalo, New York
  - Frederick the Great's death place is Indianapolis

- Pattern:
- ___ ‘s death place is ___
- Label the pattern: Subject, DIE_IN, Object

But for now...

- Let’s assume that we have the algorithm and thus we have the labeled patterns.
- Regex to the rescue!

Relation Extraction

- Some relations:
  - DIE_IN: Subject, death place is, Object
  - DIE_IN: Object, is ,Subject, last place
  - AWARD: Subject, award is, Object
  - AWARD: Object, is, Subject, honour
  - Etc.

Parse Wikipedia page

- It turns out most of the critical information about a person is contained in the infobox.
- Use Jsoup library to parse html document

Born	16 June 1723 NS (5 June 1723 OS) Kirkcaldy, Fife, Scotland
Died	17 July 1790 (aged 67) Edinburgh, Scotland
Nationality	Scottish

Occupation	Social activist, businessman, writer
Known for	Founder of the Red Cross
Children	1
Awards	Nobel Peace Prize (1901)

Comparison

- Now we can compare being more sure
  - Input: Elizabeth Taylor's death place is London
  - Triple: Elizabeth Taylor, DIE_IN, London
  - Do we have DIE_IN info from wikipedia page?
    - Yes
      - Is it the same as the input?
        - Yes -> output: 1.0
        - No -> output: -1.0
    - No -> output: 0.0

Not so many Problems left

- Not every article found on wikipedia
- Not every relation found on wikipedia
- Not every infobox on wikipedia has same structure

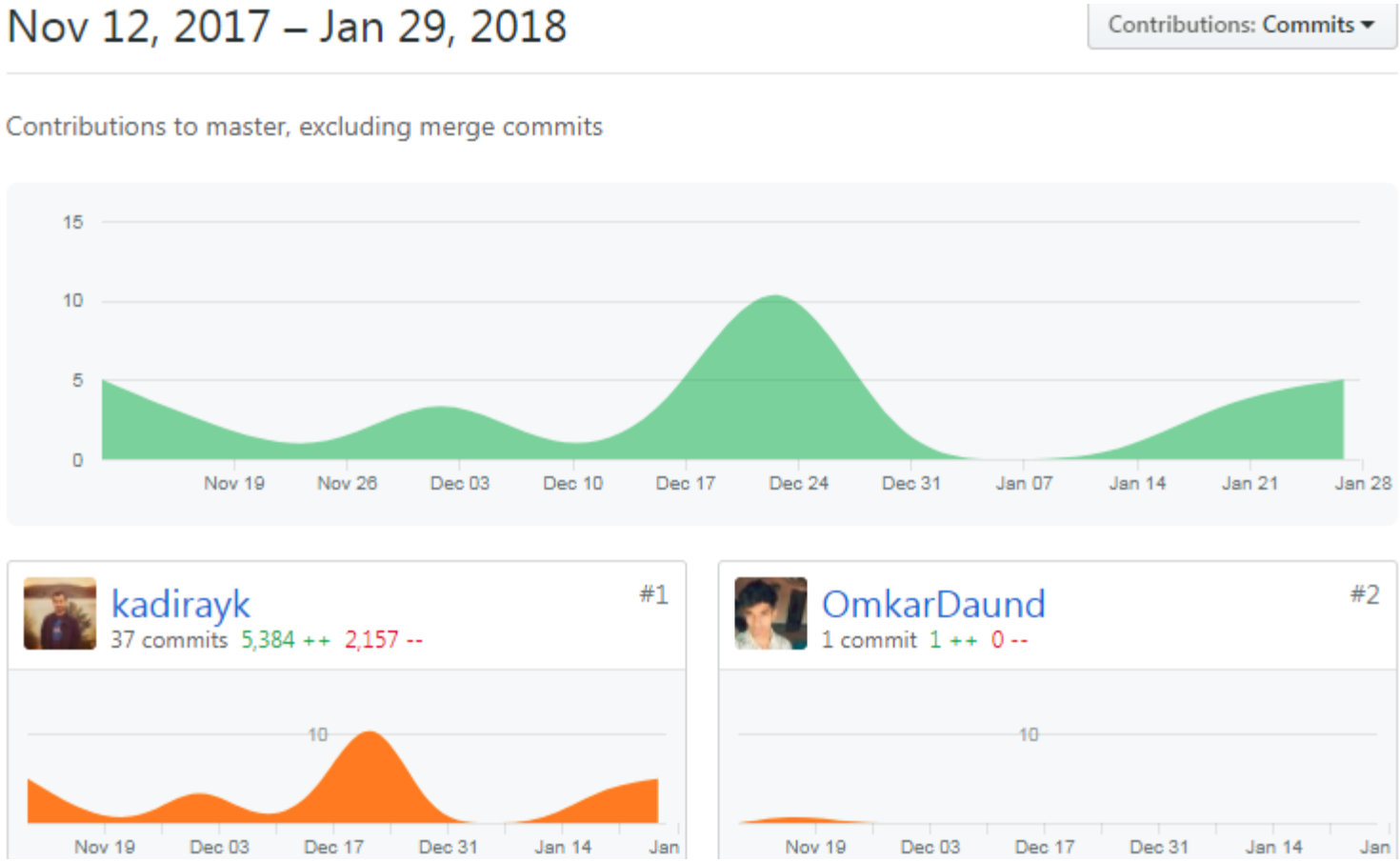
Final Thoughts

- We should replace regex part with an actual learning system.
  - Bootstrapping Information Extraction can be used
- The implementation is too spesific, it does not generalize.

Task Distribution

- Search for existing methods
- Finding Corpus
  - Parsing
  - Normalization
- Implement Stanford CoreNLP integration
- Implement Similarity measure
- Implement regex triple extraction
- Documentation

Task Distribution



- Repo: https://github.com/semicolonMissing/SNLP-Fact-Checker/