M2 Software Project

Online App for Knowledge Substantiation

Anna MOSOLOVA
Elisa LUBRINI

UNIVERSITÉ DE LORRAINE
Institut des sciences du Digital
Management & Cognition

November 22, 2021

Contents

Table of Contents

Overview

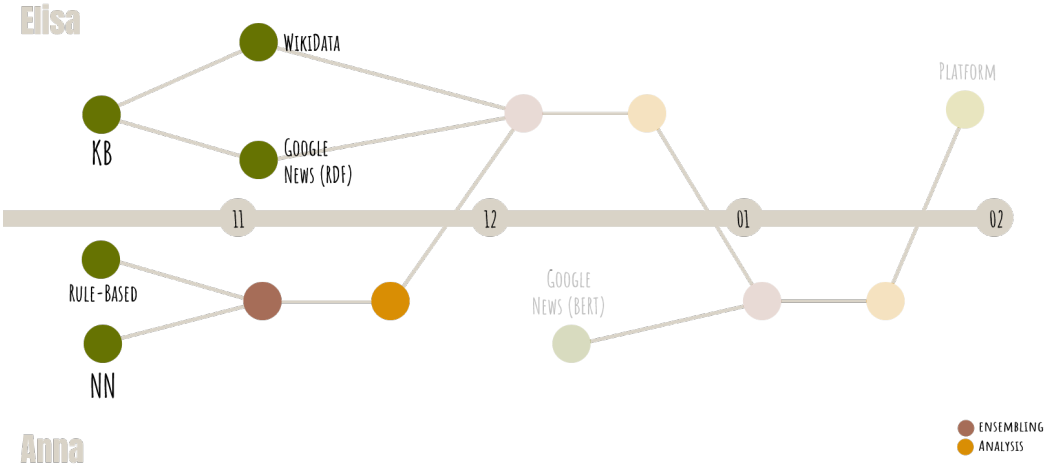

Figure: Expected Timeline

Overview

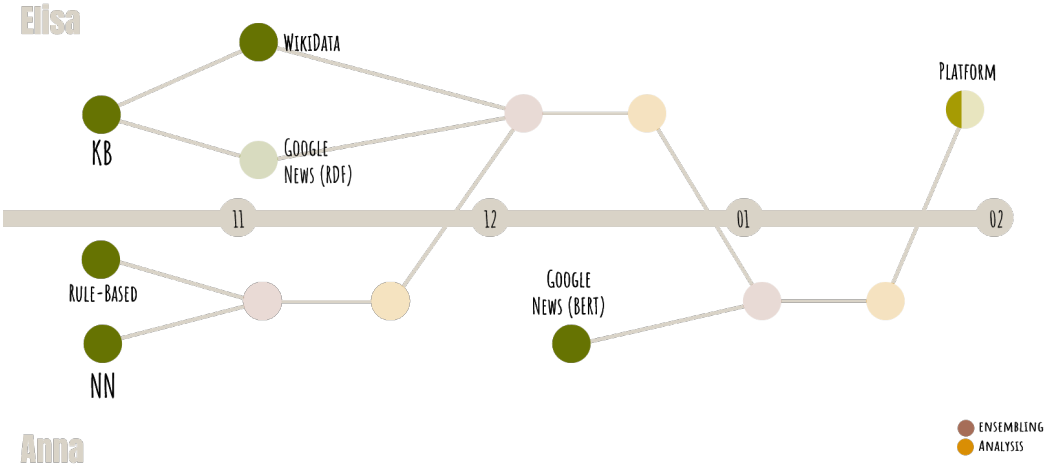

Figure: Current Timeline

Table of Contents

Knowledge Base

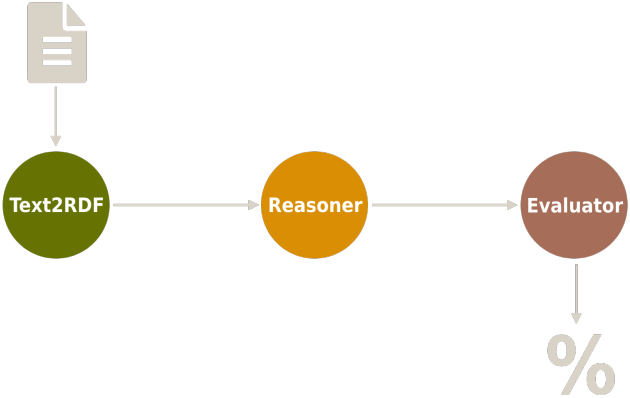

Figure: KB Pipeline

Knowledge Base

Pipeline File Structure

```
\knowledge_based_approach\
        \data\
        \output\
        \src\
                __init__.py
                info_extractor.py
                reasoner.py
                util.py
        __init__.py
        __main__.py
```

Accessing the Module from Bash

```
$ python3 knowledge_base_approach
--text filename.txt
```

Knowledge Base

Pipeline File Structure

\knowledge_based_approach
- \data\
- \output\
- \src\
  - __init__.py
  - info_extractor.py
  - reasoner.py
  - util.py
- __init__.py
- __main__.py

oaks > app > modules > knowledge_base_approach > __main__.py > ...

```python
def convert_to_rdf(text):
    from src.info_extractor import extract_information
    triples = extract_information(text)
    return triples

def check_consistency(triples, ontology, entities):
    from src.reasoner import get_accuracy
    accuracy = get_accuracy()
    return accuracy

if __name__ == '__main__':

    import argparse

    parser = argparse.ArgumentParser(description='knowledge-based substantiation')
    parser.add_argument(required=True, metavar='article',
                        help='an article to be checked')
    args = parser.parse_args()

    triples = convert_to_rdf(args)

    accuracy = check_consistency(triples)

    print(accuracy)
```

Knowledge Base

Text2RDF Tools

- × FRED → missing resource
- × Basic Custom → underperforming
- ✓ OpenIE [1] [1] → currently working, best performing script
- ? Neural Approaches → possible future implementation for improvement

[1] OpenIE python wrapper [2]: https://github.com/philipperemy/Stanford-OpenIE-Python

Knowledge Base

Reasoner

- *Output:* number of inconsistent classes
- Complexity: SPARQL is PSPACE-complete

→ import .ttl entities via Virtuoso[2]

→ use only classes without entities

[2] high-performance object-relational SQL database

Knowledge Base

Evaluator

- *Output:*
  - % of inconsistent classes/entities over total
  - name of inconsistent classes
  - [other info for algo transparency]

Table of Contents

Cross-checking Module

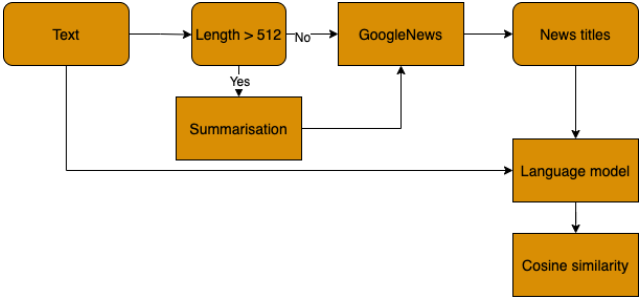

Figure: Cross-checking module pipeline

Cross-checking Module

Language model for cosine similarity

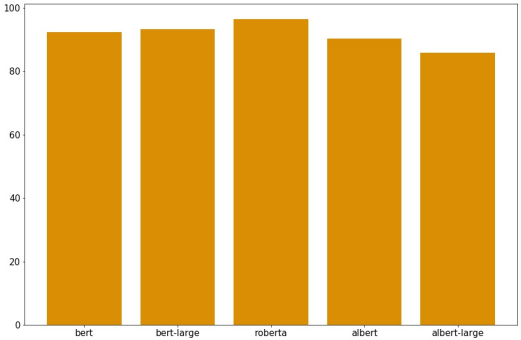

Figure: Language models comparison based on the produced similarity scores

Cross-checking Module

Summarization model

- Pegasus multinews
- **Pegasus xsum**
- BigBirdPegasus large arxiv
- BART large
- BART large CNN
- **TextRank**

Table of Contents

Next steps

Implement and try different **ensemble methods** for feature-based, NN-based and cross-checking modules

Analyse ensembling results

Add **Google News** to the KB pipeline

Implement **neural Text2RDF approach**

Load **entities from .ttl** to a graph

Bibliography I

[1] Lei Cui, Furu Wei, and Ming Zhou. “Neural Open Information Extraction”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 407–413. DOI: 10.18653/v1/P18-2065. URL: https://aclanthology.org/P18-2065.

[2] Philippe Remy. *Python wrapper for Stanford OpenIE*. https://github.com/philipperemy/Stanford-OpenIE-Python. 2020.