IN5550, Spring 2024 Home Exam
Task Description
Fact checking with graph evidence

Area chairs: Roxana Pop, Sondre Wold

Introduction

This document introduces one of the tasks for the Spring 2023 Home Exam for IN5550: **Fact checking with graph evidence**. For general instructions regarding the home exam, see the information at the semester page for the course:

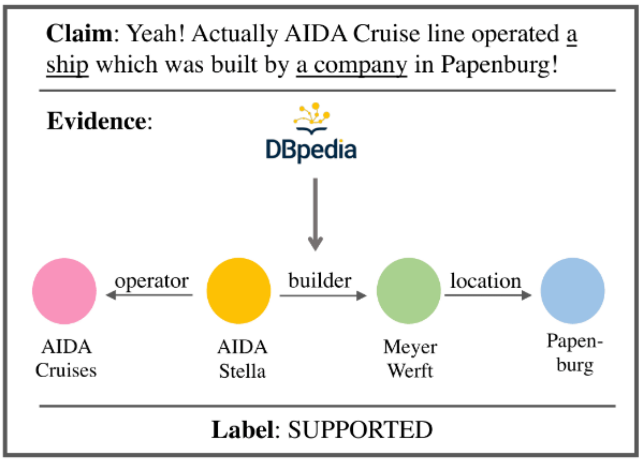
`https://www.uio.no/studier/emner/matnat/ifi/IN5550/v24/final-exam/`

Please contact the area chairs as soon as possible if you want to choose this track. There will be at least one online mentoring meeting before May 12.

The task in short

Fact checking is the NLP task of automatically verifying whether a claim is true or not, while considering a source of evidence. In this exam track you will focus on the setting introduced by Kim et al.[Kim et al., 2023], where the evidence is based on a Knowledge Graph. More concretely, given a claim with entities from DBpedia[Kobilarov et al., 2009] (a knowledge graph), your task is to predict whether the claim is *supported* or *refuted*.

Figure 1: Example of supported claim (taken from [Kim et al., 2023])



In Figure 1 we can see an example of a claim being supported by a subgraph from DBpedia. That is, the claim "Yeah! Actually AIDA Cruise linearly operated a ship which was built by a company in Papenburg!" is supported, as there is a ship (AIDA Stella), that was operated by AIDA cruises, and this ship was built in Meyer Werft, which is indeed located in Papenburg.

Dataset

You will be using the FactKG dataset introduced by Kim et al.[Kim et al., 2023]. This dataset consists of 108K claims constructed through different reasoning patterns applied on facts from DBpedia.

The original dataset is linked on Github[1], but it has been processed for easier loading and is available in *.csv* format on fox[2].

The input in all splits consists of:

- The claim (natural language sentence in English)
- The set of DBpedia entities appearing in the claim
- The label (True/False)

In the original dataset, the training and validation sets also contain the DBpedia subgraphs which can be used as evidence. More precisely, the relation paths relevant to the claim starting from each entity are known. These relation paths are predicted by the system in [Kim et al., 2023] on the inference flow.

For this exam track you can consider one of two settings:

1. Simplified setting:
   - In this setting you can consider both the claim and the subgraph evidence as input. The data for this setting was created from the training data from [Kim et al., 2023] and can be found on fox at */fp/projects01/ec30/factkg/simple*.
2. Full setting
   - In this setting you can consider that you have as input the claim and the set of DBpedia entities for the claim. If you want to use subgraph information as evidence, you must retrieve that yourself. For the training and validation data you can consider the relevant relation paths for each entitity in the claim known. This setting is that considered in [Kim et al., 2023]. You can find the data for this setting at */fp/projects01/ec30/factkg/full*.

For both these settings you can use the processed DBpedia data provided together with the FactKG dataset. This data can be found on Github[3] but has been placed on fox as well and can be found at */fp/projects01/ec30/factkg/dbpedia*. There are two versions, the complete 2015 (undirected) knowledge graph and a light version containing only the relevant edges for FactKG. We recommend

[1] https://github.com/jiho283/FactKG
[2] TODO
[3] https://github.com/jiho283/FactKG

using the light version. The data is stored in a *pickle* format and can be loaded as Python objects. Be aware that the data is quite large and needs to be loaded at once due to the format. This means you should load this data on a machine with enough memory (any of the fox GPU nodes should work, or the large fox CPU nodes). You do not need to use the data in this format and can query DBpedia through SPARQL as well, but then please make sure to indicate what version of DBpedia you queried. Also, be aware that the DBpedia released with the FactKG dataset is also processed to contain all inverse relations (in the format " relation") and you should consider this if you decide to follow the method by Kim et al. [Kim et al., 2023] while querying a different version of DBpedia.

1 Simplified setting - known evidence

In this setting you can consider you have as input:

- Claim
- Subgraph evidence

The subgraph evidence is given as relation paths starting from each entity, and relevant to the given claim.

For the example in Figure 1 the input would look as in the table bellow:

Claim	**Subgraph evidence**	**Label**
Yeah! Actually AIDA Cruise line operated a ship which was built by a company in Papenburg!	AIDA Cruises: $\sim$ *operator*, *builder*, *location*; Papenburg: $\sim$ *location*, $\sim$ *builder*, *operator*	Supported

1.1 Baseline

Two baselines are provided for this setting:

- A claim-only baseline
- An evidence-based baseline (used DBpedia)

For the claim only baseline the claim is encoded and fed to a binary classifier. For the evidence-based baseline, the paths connecting the entities are processed into text similarly to Kim et al. [Kim et al., 2023]. Once transformed into text, the paths are concatenated with the sentences of the claims to generate the input. In both cases, the textual input is encoded using a BERT-based model.

You can find more details about the baselines and how they can be run on Github.

1.2 Suggestions for improvement

You can of course experiment with different language models for processing the claim. You can also experiment with zero-shot and few-shot learning similar to what you did in the third obligatory.

Yet, we encourage you to experiment with different ways of using the graph evidence:

- You can reproduce the method introduced together with FactKG and use as evidence paths connecting the entities[Kim et al., 2023]. For this setting you do not need to predict the relevant relations for each entity: you can consider them known and you can use them together with the Knowledge graph to generate the paths.
- In path-based the method proposed by Kim et al. [Kim et al., 2023] the paths are transformed to textual data by simple processing (using special symbols to delimit the paths and their components). You can instead use a language model for generating a textual format for the paths as done by Wang et al. in a question-answering setting [Wang et al., 2020].
- You can experiment with Graph-Neural Networks for processing the subgraph evidence, for example with graph attention networks as in the QA-GNN method [Yasunaga et al., 2021] covered in the lectures.
- You can use the MHGRN module introduced by Feng et al [Feng et al., 2020] which unifies path reasoning with graph neural networks.

2 Full setting

Baseline

For this setting one baseline is provided: a claim-only method based on BERT.

Suggestions for improvement

In this setting you can consider you have as input:

- Claim
- Entity set related to claim

The output is of course still the binary label (supported or not).

For the training and validation sets you also have the graph evidence (given as paths of relations starting from each entity). You can use this data if you want to train a classifier for predicting the paths as done by Kim et al. [Kim et al., 2023]. You can also sample random walks [Wang et al., 2020] or consider all paths up to a certain length [Wang et al., 2020].

Of course, you are free to come up with your own method for selecting the relevant information from the Knowledge graph given the set of entities. You can use a combination of traversing the graph and prediction modules found in the literature or you can come up with something yourself! Just remember you also have the set of entities from the claim!

If you choose this setting than it should be possible to directly compare with the existing methods on the FactKG[Kim et al., 2023] dataset!

Good luck with your exam!

References

[Feng et al., 2020] Feng, Y., Chen, X., Lin, B. Y., Wang, P., Yan, J., and Ren, X. (2020). Scalable multi-hop relational reasoning for knowledge-aware question answering. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1295–1309, Online. Association for Computational Linguistics.

[Kim et al., 2023] Kim, J., Park, S., Kwon, Y., Jo, Y., Thorne, J., and Choi, E. (2023). FactKG: Fact verification via reasoning on knowledge graphs. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16190–16206, Toronto, Canada. Association for Computational Linguistics.

[Kobilarov et al., 2009] Kobilarov, G., Bizer, C., Auer, S., and Lehmann, J. (2009). Dbpedia - a linked data hub and data source for web applications and enterprises.

[Wang et al., 2020] Wang, P., Peng, N., Ilievski, F., Szekely, P., and Ren, X. (2020). Connecting the dots: A knowledgeable path generator for commonsense question answering. In Cohn, T., He, Y., and Liu, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4129–4140, Online. Association for Computational Linguistics.

[Yasunaga et al., 2021] Yasunaga, M., Ren, H., Bosselut, A., Liang, P., and Leskovec, J. (2021). QA-GNN: Reasoning with language models and knowledge graphs for question answering. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.