Semester Project - Fall 2021

"Language Models are Open Knowledge Graphs"
And other Methods for Relation/Entities Extraction

Student: Wafaa Tounzi
tounzi@eurecom.fr

Supervisors: Raphaël TRONCY
Ismail HARRANDO
and Thomas SCHLEIDER

Academic year: 2021/2022

Contents

1 Introduction

As the science of extracting facts from textual data has changed dramatically over the last decade and the term Natural Language Processing (NLP) took over Text Mining, the methods used have changed tremendously, too. One of the primary drivers of this variation was the emergence of language models as a basis for plenty of packages aiming to distill valuable insights from raw text.

The use of knowledge graphs which represents an assembly of interlinked entities helps with relation/entity extraction since they are considered as a form of semantic network used to map data collection from different sources and create connections between many concepts like real-world objects, events, situations or abstract concepts. They are either automatically generated or "manually".

The main focus of this work is to study the ability to extract relations between entities. We will study some methods that answer this goal starting with the idea behind "Language models are Open Knowledge Graphs" paper that aims to minimize the involvement of humans in the process of constructing knowledge graphs (KGs) from pre-trained language models. Also, we will compare their method to OpenNRE and FRED that serve the same goal.

This project is divided into four parts. We start with Relation entity extraction/linking methods from each paper. Next, we will provide an approach for entity and relation extraction by discussing the dataset used for every method. We then explain the implementation of each model. Finally, the last task would be comparing the results.

2 Related Work

2.1 Language Models are Open Knowledge Graphs paper analysis

"Language Models are Open Knowledge Graphs" paper focused on building knowledge graphs from text by leveraging Transformer based language models. Thus, this research on a high level proposes to construct knowledge graphs (KGs) from pre-trained language models using BERT, GPT-2, GPT-3 without human supervision.

Basically, the core message in this paper is that there is no training required, the entire knowledge is simply extracted from running the corpus once. So, one forward pass through the pre-trained language model and KGs are constructed to get entities and relations. Let's start by discussing the main subjects of this paper.

2.1.1 Language models

Language models determine next word probability by analyzing text data. They interpret this data by feeding it through an algorithm that establishes rules for context in natural language. Then, the model applies these rules in language tasks to accurately predict or produce new sentences. The model essentially learns the features and characteristics of a language and uses those features to understand new phrases.

The chosen pre-trained language models in this research can result in triplet classification, relation prediction and link prediction tasks. The paper provides some insights into the difference between using a BERT model vs a GPT model (auto-regressive in nature).

BERT is a state-of-the-art pre-trained contextual language representation model built on a multilayer bidirectional Transformer encoder. The Transformer encoder is based on self-attention mechanism. There are two steps in BERT framework: pre-training and fine-tuning.

As for **GPT2/3**, an autoregressive language model that uses deep learning to produce human-like text, it needs no fine-tuning and only few demonstrations to understand tasks and perform them.

The authors of "Language models are open knowledge graphs" indicate that BERT model has better success at identifying relations, compared to GPT2 and OpenIE because they are bidirectional in nature and see tokens in both directions while learning to predict masked tokens.

2.1.2 Match and Map (MAMA) Method

For extracting KGs, one of the steps was that the authors designed an unsupervised approach called "MAMA" that successfully recovers the factual knowledge inferred from Language Models to build Knowledge Graphs from scratch. MAMA constructs a KG with a single forward pass of a pre-trained LM (without fine-tuning) over a textual corpus.

To explain their MAMA method, we can simply introduce what it does this way: we have a corpus, and we want to extract relations and entities out of it with the use of Knowledge Graph. The Knowledge Graph consists of 2 distinct things, it has entities that we can think of as nouns, and relations (occupation,belonging, fatherhood, country, etc...). The purpose of any relations is connecting the entities to form meanings. The required stages to build KGs starts with Match step then Map.

Match stage: generates a set of candidate facts by matching the facts in the textual corpus with the knowledge in the pre-trained Language Model. Candidate facts are a set of extracted string triplets (namely; head, relation, and tail respectively). The longer the sentence the harder it is to extract the triplet strings.

Map stage: the candidate facts are mapped to a schema and these schemas are usually defined by humans, so in this step they were still going to rely on humans to define the schema. This stage produces an open KG via mapping the matched candidate facts from the Match stage to both fixed KG schema and open schema. If the schema of candidate facts exists in the KG schemas (Wikidata and TAC KBP) annotated by humans, we map the candidate facts directly to the fixed KG schema. Otherwise, we reserve the unmapped candidate facts in the open schema. This results in a new type of Knowledge Graph (KG), known as open KG, with a mixture of mapped facts in fixed KG schema and unmapped facts in the open schema.

2.1.3 Attention algorithm

What made their approach different was the fact they exploited the use of attention mechanisms to infer candidate triplets in text. During training, the attention weights are learned by the pre-trained transformer-based models so they can provide us with insights into the relationship between various terms in text. These weights, applied to anchor terms (noun chunks precisely) from a standard library like spaCy in their case, provide us with possible candidates' facts (head, relation, tail), which can then be disambiguated using an off-the-shelf entity linker like REL.

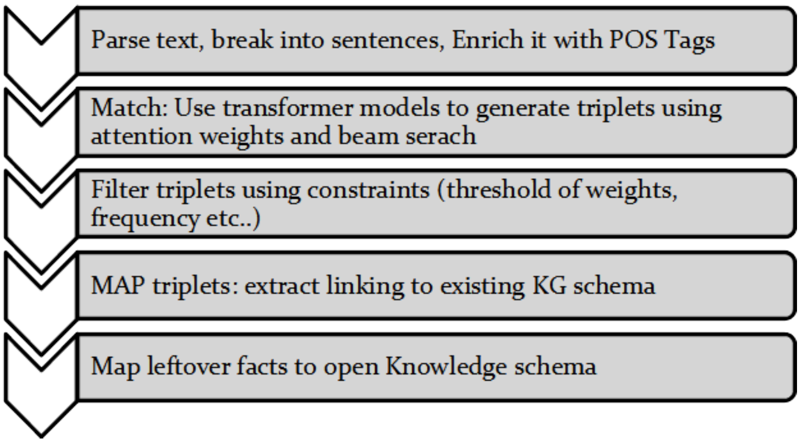
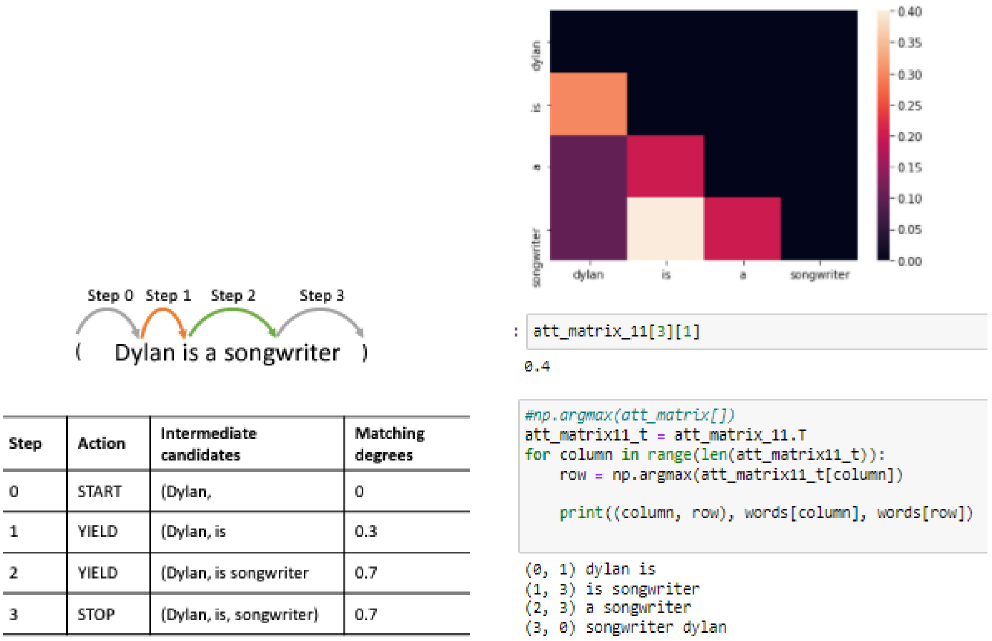


Figure 2.1 High level "MaMa" process flow proposed in the paper "Language Models are Open Knowledge Graphs"

We find the head and tail in a text, somewhere in between there might be a relation and the system needs to figure out where that is. So how does this method figure it out?

The Match process works by running a beam search to find triples with highest aggregate score. It disregards all the tokens behind the query and only looks ahead in the sentence. So that is why the upper half of the attention matrix is crossed out. From the token 'Dylan' we can only look at the tokens ahead of it (is, a, and songwriter).



Step	Action	Intermediate candidates	Matching degrees
0	START	(Dylan,	0
1	YIELD	(Dylan, is	0.3
2	YIELD	(Dylan, is songwriter	0.7
3	STOP	(Dylan, is, songwriter)	0.7

```
att_matrix_11[3][1]
0.4
```

```
#np.argmax(att_matrix[])
att_matrix11_t = att_matrix_11.T
for column in range(len(att_matrix11_t)):
    row = np.argmax(att_matrix11_t[column])

    print((column, row), words[column], words[row])
(0, 1) dylan is
(1, 3) is songwriter
(2, 3) a songwriter
(3, 0) songwriter dylan
```

Figure 2.2 Attention matrix for matching degree calculation and Matching example

2.2 OpenNRE: An Open and Extensible Toolkit for Neural Relation Extraction

OpenNRE is an open-source and extensible toolkit that provides a unified framework to implement relation extraction models. This toolkit prioritizes operational efficiency based on TensorFlow and PyTorch, which support quick model training and validation.

Since Relation Extraction (RE) aims to predict relational facts from a plain text, OpenNRE is designed for various scenarios of RE like sentence-level RE, bag-level RE, document-level RE. Based on OpenNRE architecture, the authors explain how it works to achieve its goals for system encapsulation, which is a unified underlying platform built to encapsulate various data processing and training strategies. For operational efficiency, OpenNRE is based on TensorFlow and PyTorch, which enables it to train models on GPUs. For model extensibility, various neural modules are systematically implemented and some special algorithms. Hence, we can implement new RE models based on OpenNRE.

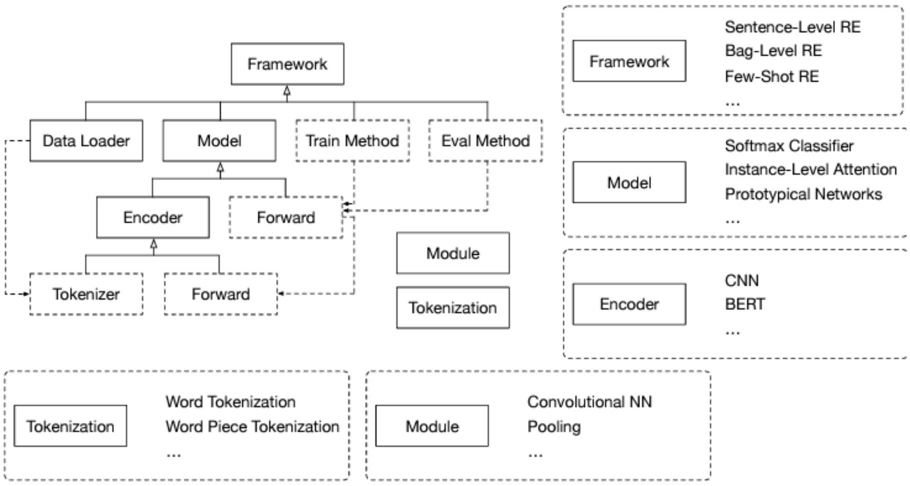


Figure 2.3 OpenNRE architecture

2.3 FRED: Semantic Web Machine Reading with FRED

FRED is a tool for automatically producing RDF/OWL ontologies (extracting RDF graphs) and linked data from text. It formally represents, integrates, improves, and links the output of several NLP tools and it has been evaluated against generic tasks (frame detection, type induction, event extraction, distant relation extraction), as well as in application tasks (semantic sentiment analysis, citation relation interpretation). Since Events are considered as exclusive entities, FRED was developed to extract them and link their type as "Event" in the graph output.

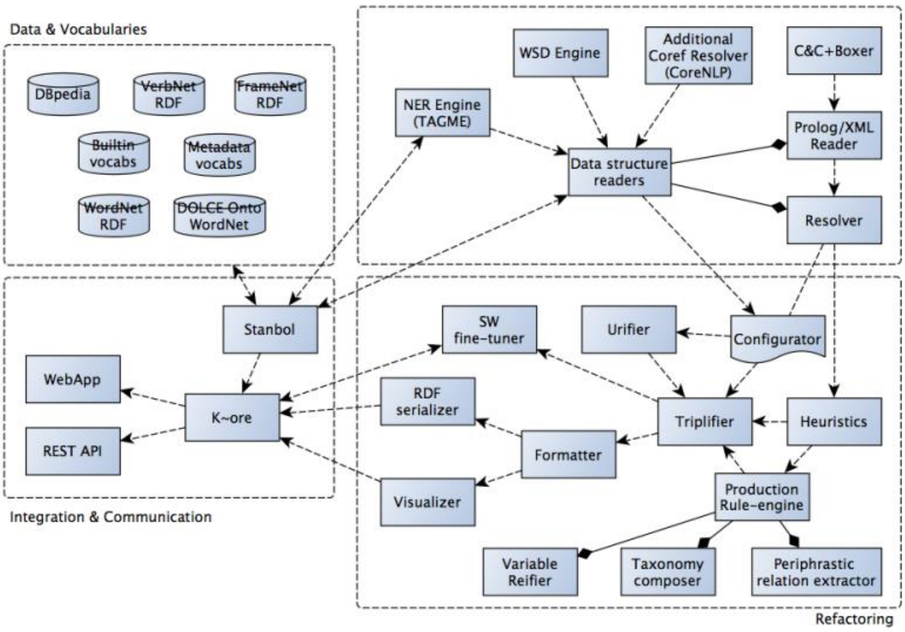


Figure 2.4 FRED architecture

The first layer takes care of reengineering, and involves external NLP components. The core of the reengineering consists of data structure readers, which read component outputs and pass them to refactoring components.

The second layer takes care of refactoring, which has its core in a modular library of heuristical rules that receive the result of reengineering readers, and produce FRED's refactoring that is sent to the triplifier component.

The third layer implements integration, communication, and alignment. Here formatted data are taken into account by a software architecture which integrates FRED with NER and WSD.

3 Approach

3.1 Dataset

We kept the same sentence “Bob Dylan is a songwriter” that the authors of “Language models are Open Knowledge Graphs” used as an input. The proposed method is unsupervised since it doesn't require any training process or data. It only requires a single application of BERT per paragraph (instead of applying BERT for every possible triplet), which is relatively efficient. But mainly, the idea of looking into the pooled attention weights for linking entities and relations is novel.

3.2 SILKNOW Knowledge Graph

SILKNOW is a research project based on records from existing catalogs that aims to produce digital modeling of weaving techniques (a “Virtual Loom”), through automatic visual recognition, advanced spatio-temporal visualization, multilingual and semantically enriched access to digital data.

SILKNOW improves the understanding, conservation and dissemination of European silk heritage from the 15th to the 19th century. By applying next-generation computing research to the needs of diverse users (museums, education, tourism, creative industries, media...), it preserves the tangible and intangible heritage associated with silk

The description of each silk heritage is based on controlled vocabularies that are essentials to link entities. The API access is developed for KG and the exploratory search engine ADASilk is created on top of it. Automatic image and text analysis are applied to predict missing metadata in the KG.

As for OpenNRE and FRED, we used four sentences from SILKNOW Knowledge Graph (full sentences in Annex section).

4 Implementation of the project

4.1 Language Models are Open Knowledge Graphs

In order to replicate the approach presented previously, we made use of the code provided by the authors **`https://github.com/theblackcat102/language-models-are-knowledge-graphs-pytorch`**. Three main steps to implement this project is to start with noun chunks for anchor terms found using en_core_web_md model from spaCy. Then, integrate Bert-large-cased model for Language Model. Last step is to use REL library for entity linking to wikidata 2019 knowledge graph.

Index of /

Name	Last modified	Size	Description
IR-Course-2021-2022/	2021-09-24 10:02	-	
SIKS-Course-2021-2022/	2021-09-30 13:48	-	
ed-wiki-2014.tar.gz	2021-03-31 14:00	1.0M	
ed-wiki-2019.tar.gz	2021-03-31 14:00	1.0M	
en-ner-fast-conll03-v0.4.pt	2021-12-16 12:02	245M	
generic.tar.gz	2020-07-07 13:18	2.9G	
ner-fast-with-lowercase.pt	2020-05-08 10:51	245M	
wiki_2014.tar.gz	2020-09-14 11:48	8.4G	
wiki_2019.tar.gz	2020-09-14 12:43	18G	

Figure 4.1 Wikidata folders.

After using their environment setup and installing the requirements, the next step is to execute "Map stage" which focuses on the entity linking using the REL repository.

REL is a modular entity linking that utilizes English Wikipedia as a knowledge base and can be used for the following tasks:

- **Entity linking (EL)**: Given a text, the system outputs a list of mention-entity pairs, where each mention is a n-gram from text and each entity is an entity in the knowledge base.

- **Entity Disambiguation (ED)**: Given a text and a list of mentions, the system assigns an entity (or NIL) to each mention.

Last step is to Execute MAMA (Match and Map) section.

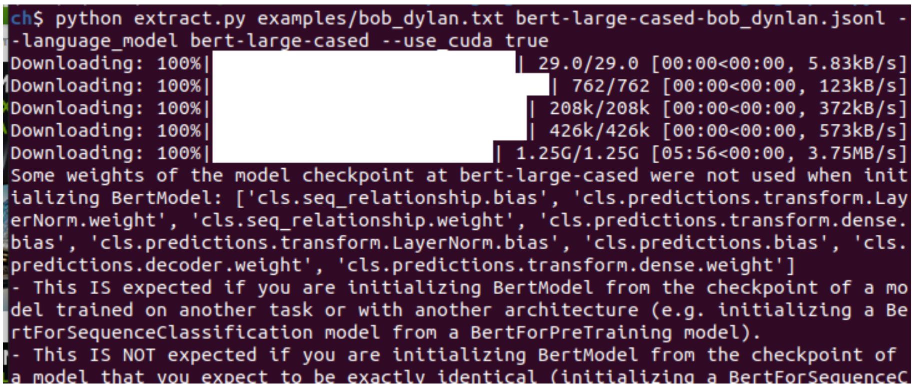


Figure 4.2 Run of command `python extract.py examples/bob_dylan.txt bert-large-cased-bob_dynlan.jsonl --language_model bert-large-cased --use_cuda true`

4.2 OpenNRE

The code is provided in this repository **`https://github.com/thunlp/OpenNRE`**. After installing OpenNRE we import the package and load pre-trained models. Then use infer to do sentence-level relation extraction. The authors lists available models like:

- `wiki80_cnn_softmax`: trained on wiki80 dataset with a CNN encoder.
- `wiki80_bert_softmax`: trained on wiki80 dataset with a BERT encoder.
- `wiki80_bertentity_softmax`: trained on wiki80 dataset with a BERT encoder (using entity representation concatenation).
- `tacred_bert_softmax`: trained on TACRED dataset with a BERT encoder.
- `tacred_bertentity_softmax`: trained on TACRED dataset with a BERT encoder (using entity representation concatenation).

Besides the toolkit, there exists an online system for OpenNRE. This online system can be directly applied for extracting structured facts from plain text. Meanwhile, all extracted entity mentions and relations can be aligned to Wikidata. The application scenarios are mainly Sentence-Level RE, Bag-Level RE, Few-Shot RE and Document RE.

4.3 FRED

FRED is available as a demo web application, and as a REST service **`http://wit.istc.cnr.it/stlab-tools/fred/fredlib.py`**. The current output of FRED is either graphic or in any RDF encoding with filtered and enriches graphs. For event classification, FRED uses Linked Data-oriented induction of types for the identified events, reusing e.g., VerbNet7, WordNet8, DBpedia9.

FRED API link is not accessible `http://wit.istc.cnr.it/stlab-tools/fred_api/`, but we can also replicate it using the Python script provided **here**.

5 Experiments Results

5.1 “Language Models are Open Knowledge Graphs” output

This schema shows what happens exactly when we create a granular version of the proposed pipeline based on their code. As we see from the flow below, the pipeline involves a lot more than just running the sentences through an LM. While an LM may help us generate candidate facts, it is equally important to tune the other parts of the pipeline. All the other pieces of the pipeline try to leverage pre-trained models from libraries like spaCy, REL...

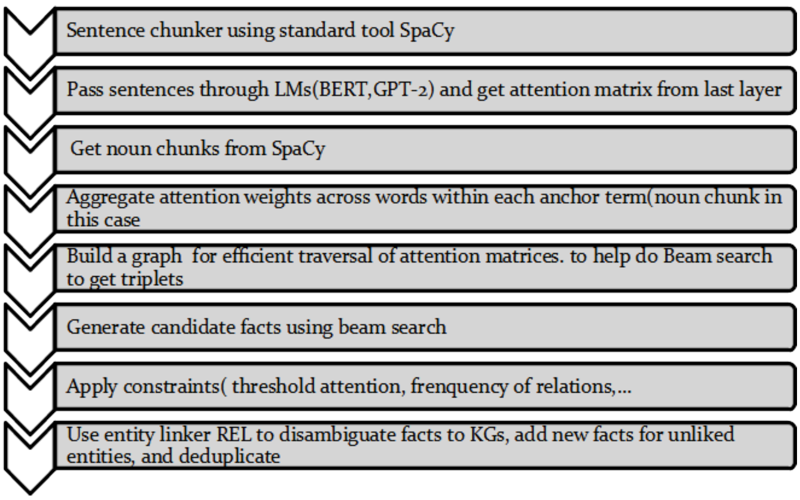


Figure 5.1 Detailed flow diagram that shows in the pipeline at a lower granularity

We get to the final result which are the KG construction of “Bob Dylan is a song-writer”:

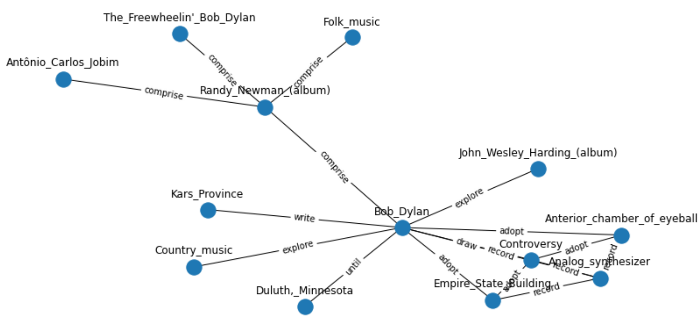


Figure 5.2 Output of the final result

- The mapped facts are existing in the final KG.
- The graphs are visually clear and understandable.
- The relation between entities is correct.
- 35.3% of the unmapped facts are verified and are true on Wikidata.

For instance, the output "Bob Dylan, signed, Albert Grossman" is an unmapped fact meaning that when the head and tail are linked but the relation between them is not mapped since there is no relation mapping from "signed" to a KG relation in Wikidata schema.

5.2 OpenNRE outputs

Using SILKNOW sentence 1 for entities extraction on the online system for OpenNRE

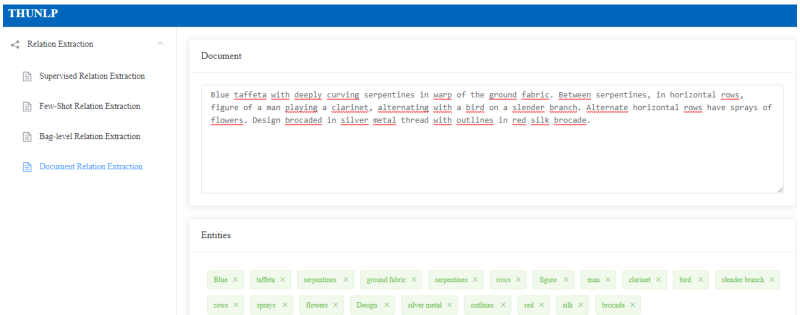


Figure 5.3 Online system for OpenNRE

- The online system has a real-time extraction without any training and deploying for every application of this toolkit, whether it's extracting entities from a sentence or a whole document.
- The API server needs maintenance and its bugs fixed. They mentioned this part as their future work.

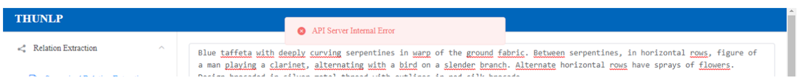


Figure 5.4 Server error

- The prediction of the relation between entities is accurate for some models like BERT and CNN
- `wiki80_bertentity_softmax` outputs most of the time "said to be same as"

```
>>> model.infer({'text': 'For Eastern Orthodox Church. Mary holding the baby Je
sus who holds a globe and makes the motion of blessing. The half figure of an a
ngel looks toward them from the upper left corner. Continuous border on the sid
es and bottom of a stylized blossom (?) repeating alternately with seraphim. To
p border pieced; a repeating botonee cross within a circle. Outer tape border.
Metal wires and silk yarns on a red satin-silk ground. Worn and missing wires;
ground worn through in some areas.', 'h': {'pos': (18, 46)}, 't': {'pos': (78,
91)}})
('said to be the same as', 0.4676766097545624)
```

Figure 5.5 Outputs of OpenNRE for sentence 2

- `tacred_bert_softmax` and `tacred_bertentity_softmax` output only “NA” each time we run the command.
- The models only predict the relation and don't add any other fact or information to the input.
- We get the same prediction but with different confidence score different models like `wiki80_cnn_softmax`, `wiki80_bert_softmax`. For instance the prediction result ‘part of’ of sentence 2, The picture is actually “part of” Eastern Orthodox Church

Model	Sentence	Relation Result	Confidence Score
`wiki80_cnn_softmax`	#1	Has part	0.3279959261417389
	#2	Part of	0.5306962728500366
	#3	Has part	0.8898584246635437
	#4	Mountain range	0.3078944981098175
`wiki80_bert_softmax`	#1	Followed by	0.4160443544387817
	#2	Part of	0.9040890336036682
	#3	Has part	0.9865756690502167
	#4	Followed by	0.8147433996200562

Table 5.1 Relation results with their confidence scores

Using `wiki80_cnn_softmax` and `wiki80_bert_softmax` models that were primarily trained on a general dataset like wikipedia, both showed accurate outputs compared to the other models. We can see by running all the sentences, some of them can get the same prediction result but still would have different scores. We notice that in the output “Part of” or “Has Part”, BERT scored the highest confidence value. We conclude that the confidence score depends a lot on the type of LM used.

5.3 FRED outputs

FRED is leveraging existing natural language methods and tools in order to obtain a unified, formalized representation of both facts and concepts expressed by a natural language text. It acts as an intermediate component for event extraction and representation, whose graphs can be combined in time series for historical tasks.

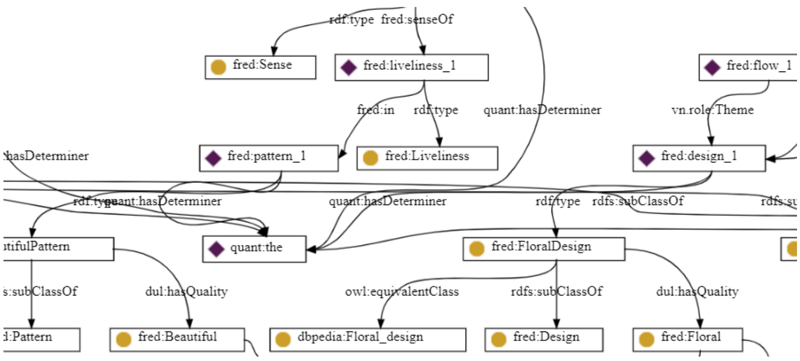
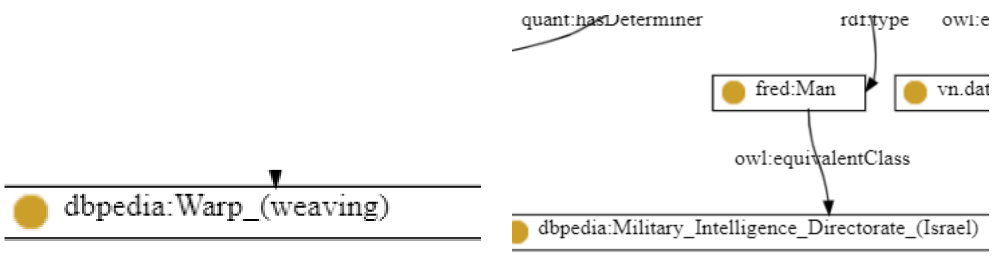


Figure 5.6 OpenNRE graph for sentence 3

The relation between every entity in the text is extracted and linked with additional facts to enrich the graph. For instance, "Warp" is extracted and linked to its quality which is "weaving".

- Visually Clear and readable Graphs
- Easy access since it's online and free
- Some outputs like "Military_Intelligence_Directorate_(Israel)" are extracted as a new fact but doesn't add any value to the input:



Linked to input entities

No useful information from sentence 1

Figure 5.7 Additional facts

- Doesn't predict relations since it only parses the text example and transforms it to linked data. The Events are extracted by using frame detection. FRED applies semantic role labeling to verbs and prepositions in order to detect event boundaries, and frame detection for resolving roles against a shared event ontology.

- In short sentences like "girl plays clarinet", FRED links Clarinet to "instrument". But for sentence 1, the word Clarinet is not linked to that.

By running all the four sentences with FRED, we conclude that for each sentence's output we would get a graph that has at least two additional facts linked to the input's entities but not all these new facts are useful. We get all the entities and the relation between them.

What to improve:

- Enriching it by combining it with other databases for more entities extractions and eventually be able to add new facts to the output
- Creating repositories of knowledge graphs that can be used to perform deep and formal annotation of large archives of documents, and to automatically produce formal relations between them.

6 Conclusion

This work is mainly based in understanding the different methods for extracting relations between entities so that we can cumulatively extract new relation facts and expand the knowledge graph as a result, which, as a way for machines to understand the human world, has many downstream applications like question answering, recommender system and search engine.

We studied from the papers how to improve the knowledge graphs with the use of language models and the power of deep machine learning using many tools. Even though the outputs are what the authors were desiring, they still had weaknesses in their program and paper:

- For "Language Models are Open Knowledge Graphs" The head and the tail entities are always single words, whereas many entities such as names have two or more words. The title of the paper is a bit strongly worded and may be overclaiming what is shown quantitatively in this paper. Their article was in-between a paper about Knowledge Base Construction (KBC) and a paper analyzing the knowledge contained within a large language model which made some confusion.

- For OpenNRE, they may improve it by adding either new facts or it can also be generating a knowledge graph based on the relation that was predicted.

- For FRED, since it's using English Wikipedia as a knowledge base, they can take it to a higher level to use other languages' data. "Knowledge Reconciliation with Graph Convolutional Networks Paper" claims that several future directions are considered, among which is the further integration of the semantics associated with knowledge graphs. That means FRED could be one of the choices for this improvement.

As future work for SILKNOW, we would like to improve its data since it needs tools that can automatically detect relations. We would like to do more research on prior work that already discussed Relation extraction with different methods or similar methods like the one using LMs to construct KGs ("Transformers for Automatic Knowledge Graph Construction." By ACL 2019 / Language Models as Knowledge Bases" by EMNLP)

Bibliography

1 C. Wang, X. Liu and D. Song, "Language Models are Open Knowledge Graphs", *CoRR* abs/2010.11967, 2020.

2 X. Han, T. Gao, Y. Yao et al., "OpenNRE: An Open and Extensible Toolkit for Neural Relation Extraction", 2019.

3 A. Gangemi, V. Presutti, D.R. Recupero et al., "Semantic Web Machine Reading with FRED", *Semantic Web* 8(6), 873-893, 2017.

Annex

All codes and repositories are provided on Github: **https://github.com/WafaaTounzi/comparison_REL**

Sentence 1: "Blue taffeta with deeply curving serpentines in warp of the ground fabric. Between serpentines, in horizontal rows, figure of a man playing a clarinet, alternating with a bird on a slender branch. Alternate horizontal rows have sprays of flowers. Design brocaded in silver metal thread with outlines in red silk brocade."

Sentence 2: "For Eastern Orthodox Church. Mary holding the baby Jesus who holds a globe and makes the motion of blessing. The half figure of an angel looks toward them from the upper left corner. Continuous border on the sides and bottom of a stylized blossom (?) repeating alternately with seraphim. Top border pieced; a repeating botonee cross within a circle. Outer tape border. Metal wires and silk yarns on a red satin-silk ground. Worn and missing wires; ground worn through in some areas."

Sentence 3: "The perfectly matched flowing floral design gives this chasuble its distinctive appearance. The sense of liveliness in the beautiful pattern comes from the finely detailed central urn motifs, which incorporate fluting and cross-hatching; the peony blossoms shaded with subtle gradations in color; and the tulip heads that are just beginning to open."

Sentence 4: "Silk, silvered-metal-strip-wrapped silk, warp-float faced satin 7:1 weave of two-color complementary ground wefts with supplementary binding warps tying supplementary brocading wefts and some self-patterning ground wefts in twill interlacing, and self-patterned by areas of plain interlacing"