KnowGL: Knowledge Generation and Linking from Text

Gaetano Rossiello*, **Md. Faisal Mahbub Chowdhury***, **Nandana Mihindukulasooriya**, **Owen Cornec**, **Alfio Massimiliano Gliozzo**

IBM Research AI
Thomas J. Watson Research Center, NY

Abstract

We propose KnowGL, a tool that allows converting text into structured relational data represented as a set of ABox assertions compliant with the TBox of a given Knowledge Graph (KG), such as Wikidata. We address this problem as a sequence generation task by leveraging pre-trained sequence-to-sequence language models, e.g. BART. Given a sentence, we fine-tune such models to detect pairs of entity mentions and jointly generate a set of facts consisting of the full set of semantic annotations for a KG, such as entity labels, entity types, and their relationships. To showcase the capabilities of our tool, we build a web application consisting of a set of UI widgets that help users to navigate through the semantic data extracted from a given input text. We make the KnowGL model available at https://huggingface.co/ibm/knowgl-large.

Introduction and Related Work

A Knowledge Graph (KG) is defined as a semantic network where entities, such as objects, events or concepts, are connected between them through relationships or properties. KGs are organized in multi-graph data structures and stored as a set of triples (or facts), i.e. (SUBJECT, RELATION, OBJECT), grounded with a given well-defined ontology (Hogan et al. 2021). The usage of formal languages to represent KGs enables unambiguous access to data and facilitates automatic reasoning capabilities that enhance downstream applications, such as analytics, knowledge discovery or recommendations (Mihindukulasooriya et al. 2022).

However, building and curating KGs, such as Wikidata (Vrandečić and Krötzsch 2014), requires a considerable human effort. Systems such as, NELL (Carlson et al. 2010), DeepDive (Niu et al. 2012), Knowledge Vault (Dong et al. 2014), DiffBot (de Sá Mesquita et al. 2019) implement Information Extraction (IE) methods for automatic knowledge base population. A standard IE pipeline consists of several steps, such as co-reference resolution (Dobrovolskii 2021), named entity recognition (Wang et al. 2021), relation extraction (Zhong and Chen 2021), and entity liking (Wu et al. 2020), each of which is commonly addressed as a separate task. A pipeline approach presents several limitations, e.g.

*Equal contributions.

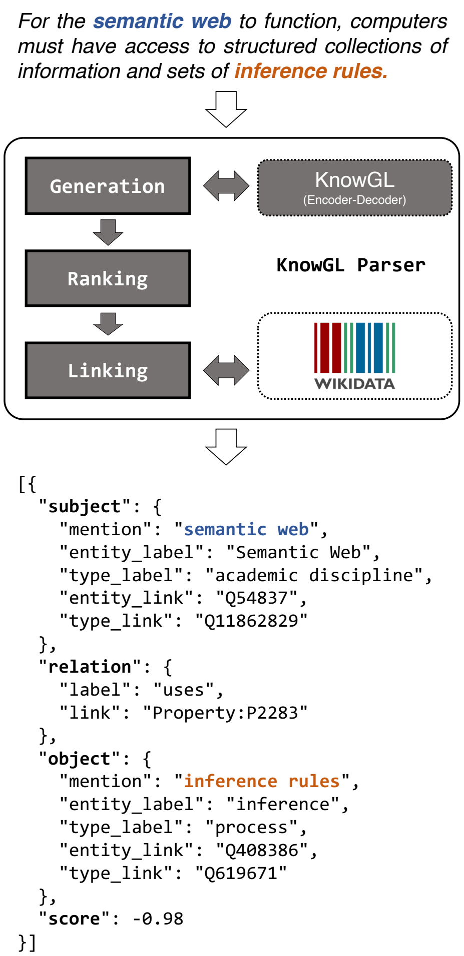

Figure 1: KnowGL Parser Framework

error propagation among different IE components and complex deployment procedures. Moreover, each component of the pipeline is trained independently using different architectures and training sets.

The ability of generating structured data from text makes sequence-to-sequence Pre-trained Language Models (PLMs), such as BART (Lewis et al. 2020) or T5 (Raffel et al. 2020), a valuable alternative to successfully address IE (Glass et al. 2021, 2022; Ni et al. 2022; Wu, Zhang, and Li 2022; Cabot and Navigli 2021; Josifoski et al. 2022), entity/relation linking (Cao et al. 2021; Rossiello et al. 2021) and semantic parsing (Zhou et al. 2021; Rongali et al. 2020; Dognin et al. 2021) tasks. In this work, we further explore this direction by asking whether PLMs can be fine-tuned to read a sentence and generate the corresponding full set of semantic annotations that are compliant with the terminology of a KG. For this purpose, we propose a framework able to convert text into a set of Wikidata statements. As shown in (Mihindukulasooriya et al. 2022), KnowGL parser can be used to automatically extract KGs from collections of documents, with the purpose to help users with semantic content exploration, to create trend analysis, to extract entity infoboxes from text, or to enhance content-based recommendation systems with semantic features.

KnowGL Parser

Figure 1 shows an overview of the Knowledge Generation and Linking (KnowGL) tool. Given a sentence as input, KnowGL returns a list of triples `(subject, relation, object)` in a JSON format [1]. From the example, KnowGL identifies the entity mentions *semantic web* and *inference rules* in the sentence and generates the relation label *use* between them. For each mention, the output provides the corresponding entity label and its type. If the entity labels, entity types and relation labels are found in Wikidata, then KnowGL also provides the Wikidata IDs associated with them. As shown in Figure 1, KnowGL Parser consists of three main components: generation, ranking and linking, as described below.

Knowledge Generation We address the fact extraction as an autoregressive generation problem. In other words, given a natural language text input, the knowledge generation model generates a linearized sequence representation that contains a set of facts expressed in the textual input. We adopt the following schema to represent the semantic annotations of a triple in the target sequence: `[(subject mention # subject label # subject type) | relation label | (object mention # object label # object type)]`. If the input text contains multiple mention pairs, the linearized target representations are concatenated using `$` as a separator, and the facts are sorted by the order of the appearances of the head entity in the input text. Unlike in (Cabot and Navigli 2021; Josifoski et al. 2022), we generate the surface forms, entity labels, and type information for both head and tail entities in the target representation. This represents a full set of semantic annotations, i.e. ABox and TBox, to construct and populate a KG with new facts. Our hypothesis is such self-contained fact representation also acts as an implicit constraint during decoding. We exploit BART-large (Lewis et al. 2020) as the base model and cast this as a translation task where at training time the encoder receives a sentence, and the decoder generates the sequence target representation as described above. To train the generation model, we extend the REBEL dataset (Cabot and Navigli 2021) by adding the entity labels and their types from Wikidata for each entity surface form in the text. REBEL is an updated and cleaner version of T-REx (ElSahar et al. 2018), a distantly supervised dataset for relation extraction built by aligning Wikipedia abstracts with Wikidata triples. We use cross-entropy loss as standard in machine translation whereas, in teacher forcing, the model regards the translation problem as a one-to-one mapping process and maximizes the log-likelihood of generating the linearized facts given the input text. As reported in (Mihindukulasooriya et al. 2022), our KnowGL model outperforms (F1 = 70.74) both a standard IE pipeline system (F1 = 42.50) and the current state-of-the-art generative IE model (F1 = 68.93) (Josifoski et al. 2022). For the evaluation, we use the test set released with the REBEL dataset.

Fact Ranking This component parses the target sequences generated by the knowledge generation model using a regular expression. The goal is to create a ranked list of distinct facts with their scores. We extract facts from all the returned sequences generated by each beam (where the number of beams is a hyper-parameter). For each extracted fact we consider the negative log-likelihood of the entire generated sequence as a score. Since the same fact can appear in different returned sequences, we sum the scores of each sequence where the fact occurs. The idea is to promote those facts/triples that occur multiple times in different beams. Finally, the facts are sorted by their scores.

Linking to Wikidata The linking component enables retrieving the Wikidata IDs associated with the generated entity, type and relation labels. For efficiency, we create label-to-IDs maps from Wikidata and store them in key-value data storage systems to avoid bottlenecks caused by running multiple SPARQL queries to a Wikidata triple store. It is worth noticing that the model can generate new entity, type, or relation labels that are not in Wikidata. In this case, the linking component returns a `null` ID and the triple can be used as a candidate for adding new facts in Wikidata.

Demonstration

KnowGL Parser is implemented in Python language and deployed as a REST API using the Flask framework. The input is a sentence and the output is a JSON format structured as shown in Figure 1. The user interface described in our video demonstration is implemented as a separate web application using Node.js and React web tool frameworks. The UI allows users to insert textual content using a textbox. Then, the returned JSON is parsed by the UI enabling different types of visualizations. For instance, the facts can be organized in a directed multi-graph where the nodes are the entities and the edges represent the relations between two entities. The user can navigate and interact with the nodes and edges to easily locate the textual evidence associated with the triples.

[1] Figure 1 shows a special case where the output consists of only one triple. However, KnowGL is able to identify multiple pairs of mentions in the input sentence and for each of them, it generates the corresponding triple with the semantic annotations in one-pass computation.

References

Cabot, P. H.; and Navigli, R. 2021. REBEL: Relation Extraction By End-to-end Language generation. In *EMNLP (Findings)*, 2370–2381. Association for Computational Linguistics.

Cao, N. D.; Izacard, G.; Riedel, S.; and Petroni, F. 2021. Autoregressive Entity Retrieval. In *ICLR*. OpenReview.net.

Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Jr., E. R. H.; and Mitchell, T. M. 2010. Toward an Architecture for Never-Ending Language Learning. In *AAAI*. AAAI Press.

de Sá Mesquita, F.; Cannaviccio, M.; Schmidek, J.; Mirza, P.; and Barbosa, D. 2019. KnowledgeNet: A Benchmark Dataset for Knowledge Base Population. In *EMNLP/IJCNLP (1)*, 749–758. Association for Computational Linguistics.

Dobrovolskii, V. 2021. Word-Level Coreference Resolution. In *EMNLP (1)*, 7670–7675. Association for Computational Linguistics.

Dognin, P. L.; Padhi, I.; Melnyk, I.; and Das, P. 2021. ReGen: Reinforcement Learning for Text and Knowledge Base Generation using Pretrained Language Models. In *EMNLP (1)*, 1084–1099. Association for Computational Linguistics.

Dong, X.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmann, T.; Sun, S.; and Zhang, W. 2014. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *KDD*, 601–610. ACM.

ElSahar, H.; Vougiouklis, P.; Remaci, A.; Gravier, C.; Hare, J. S.; Laforest, F.; and Simperl, E. 2018. T-REx: A Large Scale Alignment of Natural Language with Knowledge Base Triples. In *LREC*. European Language Resources Association (ELRA).

Glass, M. R.; Rossiello, G.; Chowdhury, M. F. M.; and Gliozzo, A. 2021. Robust Retrieval Augmented Generation for Zero-shot Slot Filling. In *EMNLP (1)*, 1939–1949. Association for Computational Linguistics.

Glass, M. R.; Rossiello, G.; Chowdhury, M. F. M.; Naik, A.; Cai, P.; and Gliozzo, A. 2022. Re2G: Retrieve, Rerank, Generate. In *NAACL-HLT*, 2701–2715. Association for Computational Linguistics.

Hogan, A.; Blomqvist, E.; Cochez, M.; d'Amato, C.; de Melo, G.; Gutiérrez, C.; Kirrane, S.; Gayo, J. E. L.; Navigli, R.; Neumaier, S.; Ngomo, A. N.; Polleres, A.; Rashid, S. M.; Rula, A.; Schmelzeisen, L.; Sequeda, J. F.; Staab, S.; and Zimmermann, A. 2021. *Knowledge Graphs*, volume 54.

Josifoski, M.; Cao, N. D.; Peyrard, M.; Petroni, F.; and West, R. 2022. GenIE: Generative Information Extraction. In *NAACL-HLT*, 4626–4643. Association for Computational Linguistics.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *ACL*, 7871–7880. Association for Computational Linguistics.

Mihindukulasooriya, N.; Sava, M.; Rossiello, G.; Chowdhury, M. F. M.; Yachbes, I.; Gidh, A.; Duckwitz, J.; Nisar, K.; Santos, M.; and Gliozzo, A. 2022. Knowledge Graph Induction enabling Recommending and Trend Analysis: A Corporate Research Community Use Case. *CoRR*, abs/2207.05188.

Ni, J.; Rossiello, G.; Gliozzo, A.; and Florian, R. 2022. A Generative Model for Relation Extraction and Classification. *CoRR*, abs/2202.13229.

Niu, F.; Zhang, C.; Ré, C.; and Shavlik, J. W. 2012. DeepDive: Web-scale Knowledge-base Construction using Statistical Learning and Inference. In *VLDS*, volume 884 of *CEUR Workshop Proceedings*, 25–28. CEUR-WS.org.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*, 21: 140:1–140:67.

Rongali, S.; Soldaini, L.; Monti, E.; and Hamza, W. 2020. Don't Parse, Generate! A Sequence to Sequence Architecture for Task-Oriented Semantic Parsing. In *WWW*, 2962–2968. ACM / IW3C2.

Rossiello, G.; Mihindukulasooriya, N.; Abdelaziz, I.; Bornea, M. A.; Gliozzo, A.; Naseem, T.; and Kapanipathi, P. 2021. Generative Relation Linking for Question Answering over Knowledge Bases. In *ISWC*, volume 12922 of *Lecture Notes in Computer Science*, 321–337. Springer.

Vrandečić, D.; and Krötzsch, M. 2014. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM*, 57(10): 78–85.

Wang, X.; Jiang, Y.; Bach, N.; Wang, T.; Huang, Z.; Huang, F.; and Tu, K. 2021. Improving Named Entity Recognition by External Context Retrieving and Cooperative Learning. In *ACL/IJCNLP (1)*, 1800–1812. Association for Computational Linguistics.

Wu, L.; Petroni, F.; Josifoski, M.; Riedel, S.; and Zettlemoyer, L. 2020. Scalable Zero-shot Entity Linking with Dense Entity Retrieval. In *EMNLP (1)*, 6397–6407. Association for Computational Linguistics.

Wu, X.; Zhang, J.; and Li, H. 2022. Text-to-Table: A New Way of Information Extraction. In *ACL (1)*, 2518–2533. Association for Computational Linguistics.

Zhong, Z.; and Chen, D. 2021. A Frustratingly Easy Approach for Entity and Relation Extraction. In *NAACL-HLT*, 50–61. Association for Computational Linguistics.

Zhou, J.; Naseem, T.; Astudillo, R. F.; Lee, Y.; Florian, R.; and Roukos, S. 2021. Structure-aware Fine-tuning of Sequence-to-sequence Transformers for Transition-based AMR Parsing. In *EMNLP (1)*, 6279–6290. Association for Computational Linguistics.