# Building High-quality Psychology Knowledge Graphs from Text using REBEL

Zeli Chen[1*], Yixuan Chen[2], Raymond Xu[3], Yao Zhao[4]

*Corresponding authors. Emails: chenzel@lafayette.edu

[1]Department of Computer Science, Lafayette College, Easton, 18042, US, chenzel@lafayette.edu

[2]The Experimental High School Attached to Beijing Normal University, Beijing, 100032, China, yixuanch2023@126.com
[3]Wayland High School, Wayland, 01778, US, Ray01778@gmail.com

[4]School of Computer Engineering, Guangzhou City University of Technology, Hubei, 510800, China, 1030072618@qq.com

**Abstract.** This paper delves into the progression of Knowledge Graphs since their inception in 1972, highlighting their applications in sectors such as healthcare and finance. We present a novel model that automates knowledge graph creation using NLP technologies like BERT, spaCy, and NLTK. The study also explores three key tool categories in knowledge graph construction: Linguistic, Autoregressive Encoder-Decoder, and BERT-based REBEL, assessing their effectiveness and adaptability. Utilizing advanced algorithms like REBEL and KG-BERT, our system expedites the training process and enhances the quality of Knowledge Graphs. Testing on 20 'Psychology'-focused Wikipedia articles, we achieve near-optimal results in just 8 epochs. The study concludes that pre-training and predictive language models hold significant promise for improving knowledge graph utility and construction.

**Keywords:** NLP (natural language processing), Text-to-Knowledge Graph, REBEL, relation extraction, psychology.

## 1 Introduction

### 1.1 History

The term “knowledge graph” was coined in 1972 by the Austrian linguist Edgar W. Schneider, in a discussion of how to build modular instructional systems for courses. Following that, in the 1980s, the first project about Knowledge Graphs, which focused on the design of semantic networks, with edges restricted to a limited set of relations to facilitate algebras on the graph appeared. In subsequent decades, the distinction between semantic networks and knowledge graphs was blurred [1].

Many topic-specific knowledge graphs were built since, such as Wordnet which captured semantic relationships between words and meanings, and Geonames capturing relationships between different geographic names and locales and associated people [2]. The first project that linked graphics and knowledge graphs was created in 1998 when Andrew Edmonds built ThinkBase which offered fuzzy-logic-based reasoning in a graphical context. In 2007, two databases,

ICBBEM 2024, March 29-31, Wuhan, People's Republic of China

DOI 10.4108/eai.29-3-2024.2347315

DBpedia and Freebase, continued down this path as knowledge repositories for general-purpose knowledge extracted either from Wikipedia or other public datasets. Interestingly, neither referred to themselves as a knowledge graph, but still developed related concepts [3,4].

Knowledge graphs (aka. Semantics network) store interlinking real-world entities – objects, events, places, people, etc. – and the relationships between them in a visual graph structure. A knowledge graph is composed of nodes (subjects or entities) and edges(relationships connecting nodes).In recent years, knowledge graphs have garnered significant attention from industries such as entertainment, finance, and healthcare in scenarios that require exploiting diverse, dynamic, large-scale collections of data [5]. Manually extracting information from thousands of documents to build knowledge graphs is rather unscalable, thus, we introduce our project – a text-to-knowledge graph model.

In our knowledge graph generator, we incorporate NLP tools such as BERT, spaCy, and NLTK to complete sentence segmentation, dependency parsing, entity recognition, etc., and overall syntax analysis, semantic analysis, and pragmatic analysis. We also make the following improvements to the existing research surrounding this topic: increasing the quality of knowledge graphs, improving word segmentation and specializing in the topic of psychology.

### 1.2 Concepts and Definitions

To clarify the transition from raw text to a Knowledge Graph (KG) is not as complex as it may seem. Neural networks, advanced mathematical models are utilized to decode the text and identify both entities and the relationships between them. The most basic units of this transition are triplets which consist of two related entities bound by a singular relation. Due to the inherent interconnectedness of the real world, one entity may be associated with multiple others creating a network of these triplets. Once these triplets are established we are effectively in possession of a Knowledge Graph. The final stage simply involves using a programming tool to visually render this graph aiding comprehension and analysis. In essence, the process boils down to a two-step transformation: first, a neural network translates the text into triplets, then through programming, we convert these triplets into a Knowledge Graph.

## 2 Three Types of Knowledge Graph Building Tools

Knowledge Graph (KG) creation involves the use of diverse tools designed to interpret and process textual data. It's like crafting an intricate web of information that machines can comprehend and making data searchable and meaningful. These tools, each with their own unique mechanics, can translate machine language to human language, then form a KG. Here we will discuss three main types of these tools: Linguistic, autoregressive encoder-decoder and BERT-related REBEL.

### 2.1 Linguistic

The first is linguistic ones such as a model called PKDE4J [6]. Song et al. 2015 introduced PKDE4J, a tool designed for public knowledge discovery, which stands out due to its exemplary efficiency and adaptability. PKDE4J takes a comprehensive approach allowing users to find a plethora of entities and identify their interconnections swiftly unlike many contemporary tools which specialize in niche domains. The tool’s real-world applications are multifarious and its flexibility is striking from assisting law firms in document analysis to aiding hospitals in patient record interpretation. PKDE4J operates in two distinct steps. First, it conducts entity extraction using a specialized 'dictionary' and then it identifies relations between the extracted entities based on established rules. A noteworthy feature is its ability to find exact matches or approximate ones akin to conventional search engines. The tool's 'dictionary' serves as a repository for specific 'entities', usually nouns that users wish to locate within the corpus of data. The extraction process is impressively swift and efficient which surpasses human speed and accuracy. Following this PKDE4J deduces the relations between the identified entities utilizing a rule-based system or a 'dependency tree-based rule'. In simple terms, it means the tool establishes a hierarchical relationship between entities akin to a family tree which aids in understanding their interdependencies. The 'rule-based system' provides a significant advantage as the users are not required to input all the rules. In PKDE4J numerous rules have already been pre-defined which presumably are based on insights gained from previous libraries and studies. While the literature does not specify the exact libraries employed it is plausible to assume that the rule development in PKDE4J may have been influenced by methodologies and approaches used in other tools such as RelEx and Befree. These tools utilize specific rule-based methods or kernels for entity and relationship extraction and these techniques may well have informed the rule creation in PKDE4J. However, it's important to note that the explicit libraries or pre-defined rule sets used in PKDE4J are not explicitly detailed in the literature. As such any assertion about specific influences remains speculative.

One example of a rule-based system is in Jiang et al.'s study, a comprehensive question-answering system is proposed [7,8]. This system merges medical expertise, knowledge graphs, and question-answering systems capable of conducting natural language-based human-machine dialogues. The system employs crawler technology to harness vertical medical websites as data sources and constructs a knowledge graph centered around diseases. This knowledge graph incorporates approximately 44,000 knowledge entities of seven types and 300,000 entities of 11 kinds and all of which are stored in the Neo4j graph database. Of particular interest for this discussion is the system's use of rule-based matching methods and string-matching algorithms to construct a domain lexicon for classifying and querying questions. This implies that the system requires rules to be manually defined or input and match the criteria for systems necessitating user-defined rules. The focus on rule-based methods highlights the ongoing relevance of rule-based systems even in an era where machine learning and automated methods are gaining prominence. This system underscores the crucial role that manually input rules can play in refining the precision of knowledge graph construction and entity extraction. It demonstrates real-world value in the medical field further cementing the importance of user-defined rules in complex and domain-specific applications. This can show how unique and special PKDE4J is, though, these are just pros and cons and it doesn't mean which one is better.

The tool utilizes machine learning algorithms to boost both its precision and adaptability, evidenced by average measures of 85% accuracy for entity extraction and 81% accuracy for relation extraction. The inner workings of PKDE4J are divided into two core modules which are the entity extraction module and the relation extraction module. The entity extraction module starts with dictionary loading followed by a series of preprocessing steps which include abbreviation replacement, tokenization, sentence splitting, parts-of-speech tagging, lemmatization, and string normalization. Following this it undertakes entity annotation using the pre-loaded dictionary and concludes with a final post-processing step. The relation extraction module on the other hand begins with a relation dictionary which helps in determining the relationship between two entities. The use case in the paper specifically focuses on the biomedical domain which classified verbs into four types which are positive, negative, neutral, and plain for better extraction. If no verb can be found, alternative words are sought which are those verb-like words that can be normalized to represent a standard verb. The final stage of this module is the extraction of rules using a Grammatical Relation (GR) tree which simplifies complex sentences into hierarchical structures (trees) for clearer understanding.

This tool employs what we call part-of-speech analysis which is basically the study of how words function in a sentence. This allows the tool to comprehend content much like how we understand language when we read or speak. It can identify whether a word is used as a noun, verb, adjective etc. which is crucial in determining its role within the data.

### 2.2 Autoregressive Encoder-decoder

The second is autoregressive encoder-decoder: This sophisticated tool operates in two stages encoding and decoding. During the encoding phase the tool receives text and tries to comprehend it, forming an internal representation which is a unique understanding that's full of numerical information. Although this may seem like an indecipherable mess to us it is the tool's way of identifying entities and relationships.

The decoding phase then follows where the tool attempts to display the previously understood entities and relationships in a way that allows us to see the connections, forming the groundwork for the KG [9]. This process is autoregressive which means the tool leverages previous data to make predictions. For example, if it's learned that "cat" is an entity and "is" indicates a relationship, it then uses this understanding to determine whether the following phrase, "an animal" is another entity or not.

This is an example of the autoregressive encoder-decoder approach. The research paper by Nayak et al. discusses an advanced method of knowledge graph construction which employs an encoder-decoder architecture [10,11]. The paper focuses on the complex challenge of extracting relation tuples which is as same as triplets from unstructured text. When these tuples appear in a block of text, they can share or overlap entities which complicates the extraction process. Historically most research has used a pipeline approach where entities are identified first followed by the discovery of relationships among them. However, this approach often overlooks the interplay among different relation tuples in a sentence, leaving potential valuable connections unrecognized. To address this Nayak et al. introduce two novel strategies. The first strategy involves a new representation scheme for relation tuples. This method allows the decoder to generate one word at a time akin to machine translation models but retains the ability to identify all tuples in a sentence, complete with full entity names of varying lengths and overlapping entities.

To reaffirm a decoder in this context acts as a translator that interprets machine-readable language into human-comprehensible text. Conventional decoders process sentences like the cat likes to eat fish sequentially - first interpreting 'the cat' then 'likes' followed by 'to eat fish'. However, the novel model presented in the research paper introduces a decoder capable of handling a group of triplets at once. Take for example a sentence that reads "Bob likes football and Bob likes basketball." A traditional decoder after processing the first triplet "Bob likes football" forgets about it when it proceeds to "Bob likes basketball." This conventional method lacks continuity and the ability to associate information from previously processed triplets. In contrast the innovative decoder proposed in the study processes all the triplets from the provided content simultaneously while maintaining a memory of the triplets it has already interpreted. As a result this decoder is adept at summarizing that Bob has a profound interest in sports drawing from the related triplets "Bob likes football" and "Bob likes basketball". It excels in its ability to analyze multiple triplets sharing the same entity, painting a more comprehensive and contextual picture than the standard decoder.

The second strategy employs a pointer network-based decoding approach where an entire tuple is generated at every step. This method seems especially adept at handling the intricate task of relation extraction. Testing these new methods on the publicly available New York Times corpus showed promising results. Both proposed approaches outperformed previous methods and achieved significantly higher F1 scores which is an indicator of precision and recall.

### 2.3 BERT-related REBEL

The third is BERT-related REBEL: BERT is an absolute maestro when it comes to understanding the context within a text. The REBEL tool built on BERT is incredibly efficient in inferring meanings based on the context, making it an integral part of KG building. Like a seasoned detective, it can make out hidden connections between words, making our digital data web more intricate and meaningful.

## 3 Dataset

The datasets we used and their descriptions (see Table 1).

**Table 1.** Datasets and Descriptions.

| Dataset | Description |
|---|---|
| CONLLO4 | CONLL04 (Roth and Yih,2004) is composed of sentences from news articles, annotated with four entity types (person, organization, location and other and five relation type (kill, work for, organization based in, live in and located in). |
| DocRED | DocRED (Yao et al. 2019) is a recent dataset created similarly to our pre-training data, by leveraging Wikipedia and Wikidata. |
| NYT | NYT (Riedel et al. 2010) in a dataset consisting of essences from the New York Times corpus |
| ADE | ADE (Guingappa et al. 2012) is a dataset in the bical domat for which Advate-Effects from drugs are mutated as pairs of drug and adverse-effect. The dataset provides 10-folds of train and test splits |

**Table 2.** Datasets and statistics.

| Dataset | Entity Types | Relation Types | Train | | Validation | | Test | |
|---|---|---|---|---|---|---|---|---|
| CONLLO4 | 4 | 5 | 1.290 | (922) | 343 | (231) | 422 | (288) |
| DocRED | 6 | 96 | 3.7486 | (3.008) | 3.678 | (300) | 8.787 | (700) |
| NYT | 3 | 24 | 94.222 | (56.196) | 8.489 | (5.000) | 8.616 | (5,000) |
| ADE | 2 | 1 | 6.821 | (4.272) | | | | |

Here is the Number of triplets with the number of instances in parenthesis (see Table 2).

## 4. Metrics

**Table 3.** Results of our training.

| Dataset | Model | Precision | Recall | F1 |
|---|---|---|---|---|
| NYT | REBEL1 | 90.26 | 91.50 | 90.88 |
| | REBEL2 | 91.50 | 92.02 | 91.76 |
| | REBELpre - training | 91.71 | 92.21 | 91.96 |

We trained the REBEL1 for 8 epochs. REBEL2 for 42 epochs. REBELpre−training for 3 epochs (see Table 3). We can see that REBEL2 is the final performance of the model under the NYT, but it has been trained for at least 42 epoches, while the pre-trained model has been trained for only 3 epoches to almost reach the final performance, and the model we trained has only been trained for 8 epoches, which is very close to the final performance. This not only shows the high training efficiency of the model, but also reflects the importance of pre-training.

## 5. Model & Methodology

### 5.1 Flowchart

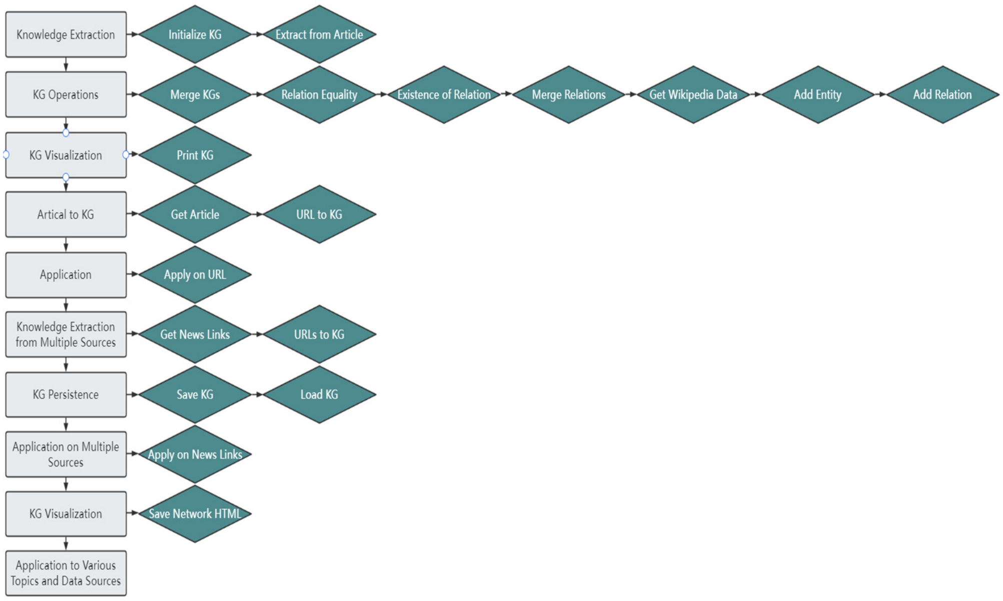


**Fig. 1.** Flow chart of Knowledge Graph construction [12].

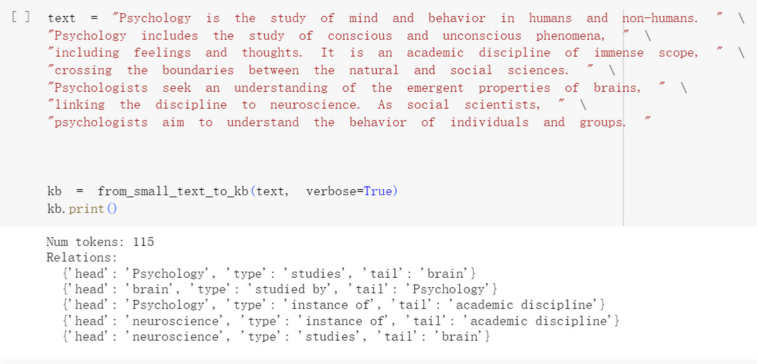
The model under consideration as outlined in the flowchart (see figure 1) and its corresponding annotations hinges upon the construct of a Knowledge Base (KB) which in this context is equivalent to a Knowledge Graph (KG). The preliminary portion of the system is intrinsically concerned with the extraction of knowledge. The function entitled 'Extract from Article' is devised to metamorphose a raw web article into a structured knowledge base. This transformation is achieved through a methodical sequence of procedures including the disassembly of complex information into cognitively manageable tokens, the delineation of precise boundaries for these tokens, the strategic transmutation of the input, the generation of meaningful relations, the decoding of intricate predictions and ultimately, the construction of the ensuing knowledge base. To maintain control over these extensive operations the function 'Initialize KB' is introduced. This function rigorously defines the KB class and instigates the arrangement of its principal components, consisting of entities, relations and sources.

Transitioning to the second facet of the system the focus intensifies on the administration and operation of the knowledge graph (KG). This segment is equipped with a suite of functions specifically engineered to enable the consolidation of two distinct knowledge bases. Alongside this it undertakes the obligation of scrutinizing and affirming the equality between two diverse relations, confirming the existence of a designated relation within the system's framework and merging two congruous relations into one entity. To enhance its functional capabilities this segment is further enriched with functions that permit the extraction of pertinent data from Wikipedia for a specified entity and the addition of novel entities or relations to the knowledge base.

The third sector of the system lays emphasis on the graphical representation and metamorphosis of the Knowledge Graph (KG). The 'Print KG' function serves a pivotal role by rendering the information contained within the knowledge base including entities, relations and sources more easily consumable through its structured display capabilities. Concurrently the function named 'Get Article' is equipped to procure an article from any specified URL and the 'URL to KG' function takes up the mantle of transmuting this acquired article into a structured knowledge base. Moving forward to the fourth stratum of the system it is devoted to illustrating the interactions of these functions across a multitude of information sources. Encapsulated within this segment is the 'Get News Links' function meticulously designed to acquire news article URLs that are relevant to a designated query, utilizing Google News as its primary information source. Subsequently the 'URLs to KG' function embarks on the task of converting these procured articles from varying URLs into a singular, integrated knowledge base. In its fifth segment the system integrates the capability to maintain persistence of the knowledge base. This implies the ability to save the KB into a file for future reference and also to retrieve it from said file when necessary. The functions namely 'Save KG' and 'Load KG' spearhead this accomplishment, providing a seamless experience of storage and retrieval. The final division of the system flaunts its versatile nature exhibiting the application of these functions across a diverse array of topics and assorted types of sources. The function labelled 'Save Network HTML' transfigures the knowledge base into an interactive visualization within an HTML format. This graphical representation manifests itself in the form of a network where nodes symbolize entities and the connecting edges denote relations.

### 5.2 REBEL

We can parse the strings generated by REBEL and transform them into relation triplets [13].



```
[ ] text = "Psychology is the study of mind and behavior in humans and non-humans. " \
    "Psychology includes the study of conscious and unconscious phenomena, " \
    "including feelings and thoughts. It is an academic discipline of immense scope, " \
    "crossing the boundaries between the natural and social sciences. " \
    "Psychologists seek an understanding of the emergent properties of brains, " \
    "linking the discipline to neuroscience. As social scientists, " \
    "psychologists aim to understand the behavior of individuals and groups. "



    kb = from_small_text_to_kb(text, verbose=True)
    kb.print()

    Num tokens: 115
    Relations:
      {'head': 'Psychology', 'type': 'studies', 'tail': 'brain'}
      {'head': 'brain', 'type': 'studied by', 'tail': 'Psychology'}
      {'head': 'Psychology', 'type': 'instance of', 'tail': 'academic discipline'}
      {'head': 'neuroscience', 'type': 'instance of', 'tail': 'academic discipline'}
      {'head': 'neuroscience', 'type': 'studies', 'tail': 'brain'}
```

**Fig. 2.** Parsing the REBEL strings (owner draw).

We try extracting a knowledge base from twenty articles from Wikipedia about “Psychology”. Then, we visualized the output of our work by plotting the knowledge bases (see Figure 2). As our knowledge bases are graphs, we can use the pyvis library, which allows the creation of interactive network visualizations. This is the resulting graph (see Figure 3 and 4).

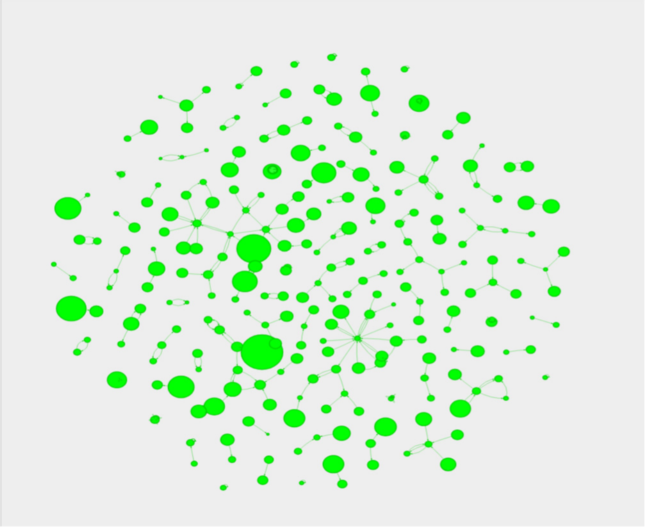


**Fig. 3.** Knowledge Graph about Psychology (owner draw).

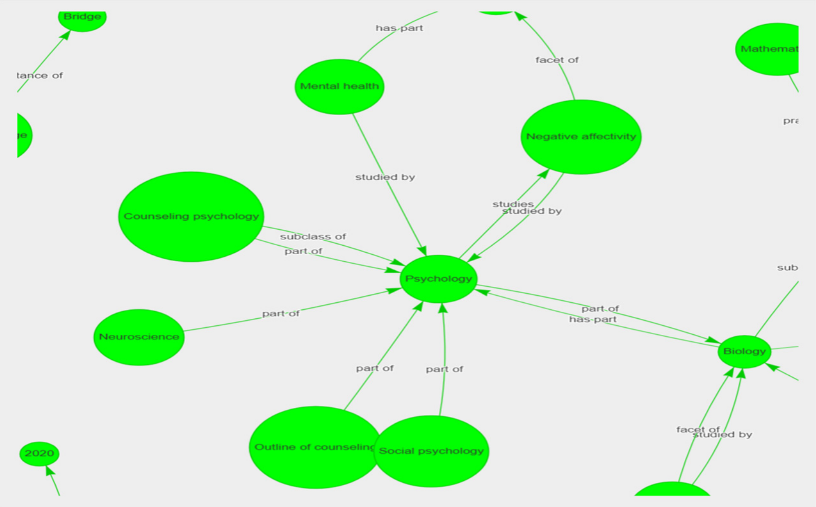


**Fig. 4.** Zoomed in section of our graph (owner draw).

## 6. Conclusion

Chen et al.'s paper provides a comprehensive exploration of the status and trends of knowledge graph research focusing on thematic structures. The authors analyzed 386 research articles published between 1991 to 2020 studying trends in the annual article and citation counts, major institutions, countries/regions, publication sources, and scientific collaborations. They identified key research themes and their developmental tendencies. The study highlights China as a major contributor to knowledge graph research and the importance of international collaboration for impactful research. The study also provides future directions and suggestions for knowledge graph research but does not mention specific breakthrough models or advancements. Looking ahead once the groundwork by constructing the basic Knowledge Graph (KG) is laid, the challenge doesn't end. The next critical stride involves refining this KG to enhance its quality and utility. Yao et al. proposes a novel model, KG-BERT, for improving knowledge for graphs [14]. Knowledge graphs often suffer from missing information which necessitates a process called knowledge graph completion to predict and fill in these gaps. Knowledge graph completion addressed by KG-BERT acts as a vital lynchpin not only filling the existing gaps with predictive prowess but also enhancing the precision, reliability and overall utility of the knowledge graph thereby transforming it into a more robust tool for information representation and extraction.

This tool exhibits its prowess by completing a KG that's already been built around triplets. Essentially it works to fill in the gaps, fortifying the existing KG. Again in the context of knowledge graphs a "triplet" is a set of three linked elements consisting of a head entity, a relation, and a tail entity which together form a fact or a piece of information. The KG-BERT model treats these triplets as sequences of text and uses a pre-trained language model specifically BERT to predict the plausibility of a triplet or a relation. The process is asking the model to read a sentence describing entities and their relations and then predict whether this sentence (or triplet) might be true. To achieve this they fine-tune the BERT model to understand and interpret these sentences and then use it to predict the accuracy of triplets in various knowledge graph completion tasks. Their experimental results indicate that this model delivers excellent performance surpassing previous state-of-the-art results on several benchmark datasets. Furthermore, the authors propose potential future directions for improvement such as jointly modeling textual information with the structure of knowledge graphs or employing more advanced pre-trained models like XLNet. This study provides an intriguing perspective and innovative approach to the challenging problem of knowledge graph completion using advanced language models.

## References

[1] Chen, Z., Wang, Y., Zhao, B., Cheng, J., Zhao, X., & Duan, Z. (2020). Knowledge Graph Completion: A Review. IEEE Access, 8, 192435-192456. https://doi.org/10.1109/ACCESS.2020.3030076

[2] Fellbaum, C. (2010). WordNet. In Theory and Applications of Ontology: Computer. ISBN: 978-90-481-8846-8.

[3] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2007). DBpedia: A Nucleus for a Web of Open Data. In Aberer, K., et al. (Eds.), The Semantic Web. ISWC ASWC 2007. Lecture Notes in Computer Science, 4825. Springer. https://doi.org/10.1007/978-3-540-76298-0_52

[4] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. Proceedings of the Association for Computing Machinery (ACM), 1247–1250. 14https://doi.org/10.1145/1376616.1376746

[5] Wikipedia contributors. https://en.wikipedia.org/wiki/Knowledge_graph

[6] Song, M., et al. (2015). PKDE4J: Entity and Relation Extraction for Public Knowledge Discovery. Journal of Biomedical Informatics, 57, 320-332.

[7] Jiang, Z., Chi, C., & Zhan, Y. (2021). Research on Medical Question Answering System Based on Knowledge Graph. IEEE Access, 9, 21094-21101. https://doi.org/10.1109/ACCESS.2021.3055371

[8] Chen, X., et al. (2021). Topic Analysis and Development in Knowledge Graph Research: A Bibliometric Review on Three Decades. Neurocomputing, 461, 497-515.

[9] Yao, L., Mao, C., & Luo, Y. (2019). KG-BERT: BERT for Knowledge Graph Completion. Northwestern University.

[10] Glavas, G., Nanni, F., & Ponzetto, S. P. (2016). Unsupervised Text Segmentation Using Semantic Relatedness Graphs. Proceedings of the Association for Computational Linguistics (ACL).

[11] Nayak, T., & Ng, H. T. (2019). Effective Modeling of Encoder-Decoder Architecture for Joint Entity and Relation Extraction. arXiv:1911.09886 [cs.CL].

[12] Chiusano, F. (2022). Building a Knowledge Base from Texts: A Full Practical Example. Retrieved from https://medium.com/nlplanet/building-a-knowledge-base-from-texts-a-full-practical-example-8dbbffb912fa

[13] Huguet Cabot, P. L., & Navigli, R. (2021). REBEL: Relation Extraction By End-to-end Language Generation. In Findings of the Association for Computational Linguistics: EMNLP 2021 (pp. 2370–2381). Association for Computational Linguistics.

[14] Yao, L., Mao, C., & Luo, Y. (2019). KG-BERT: BERT for Knowledge Graph Completion. Northwestern University.