FEDERAL STATE AUTONOMOUS EDUCATIONAL INSITUTION FOR HIGHER PROFESSIONAL EDUCATION NATIONAL RESEARCH UNIVERSITY
«HIGHER SCHOOL OF ECONOMICS»
Faculty of Computer Science

Artem Aidinian

Построение графов знаний из текстов

Building knowledge graphs from texts

Qualification paper – Master of Science Dissertation
Field of study 01.04.02 «Applied Mathematics and Informatics»
Program: Data Science

Student
Artem Aidinian

Supervisor
Michael Zakharyaschev

Moscow, 2022

Abstract

This thesis proposes a pipeline for an automatic construction of knowledge graphs from raw texts based on an artificial neural network model called $mbBARTlarge-50$. We also review state-of-the-art methods for solving a task of relation extraction as well as explore various datasets and approaches to collect them. A key contribution of this paper will be an end-to-end pipeline for building knowledge graphs from texts, a dataset for pretraining relation extraction models and an academic research.

Keywords: [nlp, knowledge graphs, relation extraction, transformers]

Contents

1. Introduction

A knowledge graph (KG), also known as a semantic network, is a data structure concept that represents real-world entities and their relations in one network. Such networks find their application in different industries: ads, healthcare, retail and finances. KGs are known to be used, for example, by Google as an information system for their Google Search[1].

Various sources[2] propose methods for building KGs from structured data or knowledge bases such as DBPedia, Wikidata, or discontinued Freebase. These methods are relying on limited information that may not fully cover the entire domain of interest. Also, they are unable to process data that was not structured by a human or an upstream system. In order to achieve better data coverage, we have no choice, but to use a system that will be able to process large unstructured sets of texts. Given that, this work will primarily focus on building KGs from raw texts. Current study[3] shows that building KGs requires solving such a task as relation extraction (RE). The goal of RE is to extract entity-relation-entity triples (e.g. node-edge-node of a final graph) from texts. A vast majority of the latest research on relation extraction utilizes language models based on Transformer architecture - a new neural network that uses the Attention mechanism to learn relations between words.

With that said, the goal of this work is to explore the concept of the knowledge graph and the modern Transformer architecture as well as dig into the current study of relation extraction task and propose a pipeline for building knowledge graphs from raw texts.

We show that KGs have several advantages compared to classical relational databases and demonstrate how Transformers can be employed to extract relations that are used to construct the knowledge graph. We will also see how different ap-

proaches to relation extraction work and explain why old sequence tagging techniques are outperformed by modern marker-based ones. Also, we will contribute the dataset for the relation extraction task that was made with a tool that utilizes machine learning to automatically generate training samples from Wikidata and Wikipedia. In the end, we propose a pipeline that incorporates a pre-trained marker-based model to construct a knowledge graph.

This work is divided into several sections: Section 2 provides a brief report of a theory that lies beneath KGs, Section 3 describes language models based on Transformers, Section 4 investigates the problem of relation extraction, Section 5 reviews state-of-the-art methods for solving RE task and Section 6 gives a pipeline for KG construction with instruction on the training process. The last sections, 7 and 8, describe Future Work and Conclusion correspondingly.

2. Knowledge Graph

KGs have become a widely used concept for storing structured knowledge. It was first introduced by Google as a part of Google Search engine in 2012 and contained facts from the CIA World Fact Book and Wikipedia. Then, KGs quickly became popular among other technology companies including Amazon, Facebook, and Microsoft. There are also non-commercial companies like Wikipedia which hosts projects like Wikidata and DBpedia. Projects that are part of crowdsource initiative like ConceptNet order collaboratively built KGs comprising world facts. Natural Language Processing and Computer Vision are used to extract information that is used to build KGs.

This chapter describes the concept of knowledge graphs. We will define what a typical KG comprises and explore general ideas on how to create such a graph.

2.1 Basic structure blocks

As with any other graphs, KGs comprise nodes and edges which are commonly called *entities* and *relations*[4]. Entities and relations are labelled, usually, with strings, numbers or IRIs (internationalized resource identifiers). These labels serve as identifiers.

Definition 2.1.1. An entity is a node in a knowledge graph that represents an instance of a given domain, e.g. politician in political news or a name of plant part in botanical readings.

Definition 2.1.2. A relation is a directed connection between two entities.

For example, if we have nodes A and C and their relation B they would form a triple (A, B, C), see Figure 2.1. We see that edge B has direction. It means that the

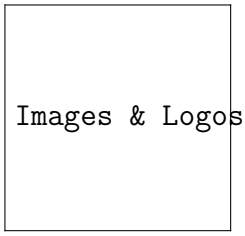

Figure 1: triple in a knowledge graph.[4].

knowledge graph is also directed.

In the (A, B, C) triple, A is a subject entity and C is an object entity. These definitions are also replaced with head or tail respectively.

2.2 Knowledge graph as a data storing structure

One of the most common ways to store data is to use a relational database with tables and a predefined set of relationships between them. Moreover, each table has a strict *schema* (e.g. set of column data types). Since the format is predefined, data has to fit it to be "processable". This strictness allows to apply databases for operational and analytical purposes, but it limits rearrangements. The more tables and rows in relational databases the more time it takes to execute the query.

On the other hand, knowledge graphs allow to store data sources with multiple relationships across nodes and add more nodes without having to comply with the schema. Knowledge graph as a graph database can run much faster on "bigger" data and allows easier maintenance due to schema-free storage which does not require analyzing and understanding the entire logic of the database. Nevertheless, KGs are

less suitable for operational purposes.

2.3 Applications of knowledge graphs

As seen above, KGs are used to enhance web search and provide brief information for certain queries. But the application is not limited to search. Knowledge graphs find their place in finding insights that cannot be described by relational databases.

Stanford AI Lab[4] provides an example of a potential application. Financial institutions are interested in combining internal data about their clients and publicly available data like news to provide an individual approach for their customers. This is called 360-degree view of a customer. Let us provide an example. If company A has a supplier B who is reported to undergo a bankruptcy process, it is worth notifying company A about new risks. See this example in Figure 2.3.

Other applications can be found in Question Answering or Recommender systems. Social Networks can apply KGs to measure privacy disclosure or to perform fake news detection. The list of applications continues to grow.

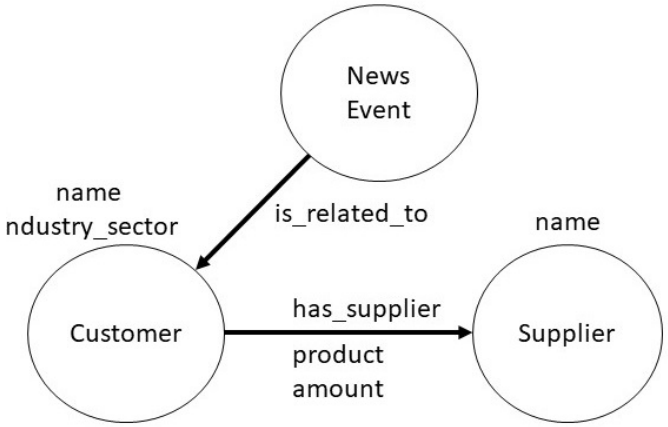

Figure 2: Example of a 360-degree view graph.

2.4 Construction of Knowledge Graph

KGs are constructed in different ways. One can use structured data, for example, personal pages of a social network to construct a knowledge graph and explore the interests of a certain group of people. Semi-structured data can be also used. Parsers for Wikipedia articles can extract facts from templates and infoboxes, see Figure2.4. Unstructured texts may populate Knowledge graphs using a human annotation or an artificial intelligence.

One of the typical public KGs is Wikidata. It is used to populate aforementioned infoboxes on Wikipedia pages. Google search uses this information to enhance search results, see Figure 4. A query *"wikipedia Einstein"* in Google outputs a link to a Wikipedia article about Albert Einstein and provides brief facts containing *"Education","Citizenship","Birth date"*, and more. An example of a knowledge graph created by means of AI can be seen in Figure 5. We see that a small text about the

Figure 3: Quick facts from infobox on Wikipedia about Albert Einstein.

famous physician can be analyzed to create a detailed KG.

Structured data is more capable of the easy creation of KGs because we are not burdened by having to process natural texts to extract entities and relations. On the other hand, the main limitation of using structured data is that it is not always covering the entire domain of interest. We need a system which that be capable of identifying entities and relations and extracting them to populate the KG. Recent studies showed that an automatic relation extraction (RE) task relies on a machine learning architecture called Transformer. Before reviewing papers on RE, we need to describe Transformer and its place in Natural Language Processing.

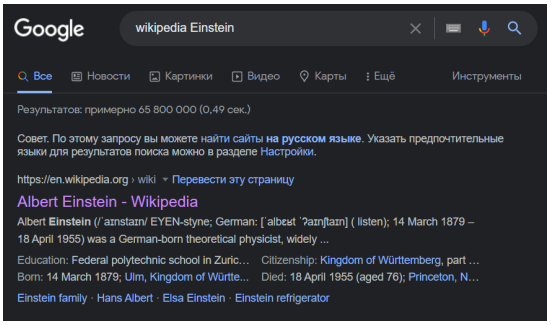

Figure 4: Example of "Wikipedia Einstein" query from Google Search

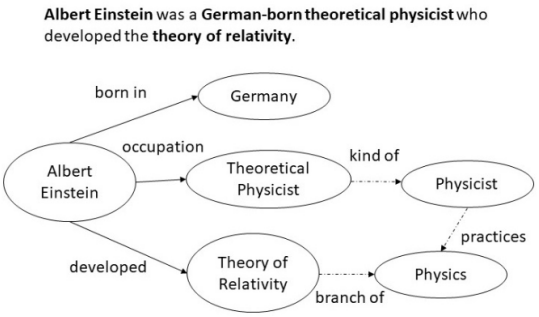

Figure 5: Example of knowledge graph describing Albert Einstein[4]

3. Language Models

We explored what Knowledge graph is and how it can be applied in industrial tasks. A core concept in our work of constructing KG from texts will be Language Model. Recent years showed noticeable growth of works dedicated to language modelling with a develop of numerous multi-modal models. They key part of these models is an Attention mechanism which became part of famous Transformer model and its descendants. We will explore what Attention is and how it changed the area of NLP and also show how transformers can be applied in our work

3.1 Sequence models

First of all, we need to start here with a brief overview of a predecessor of the current widely used Transformer architecture. We will provide a shallow explanation of RNN, LSTM and GRU neural networks that process sequential data and then switch to a backbone of many state-of-the-art - Transformers.

Text is one form of sequential data. Standard models that process tabular data cannot maintain texts because their length is variable. In the field of neural networks, such a model as a **Recurrent Neural Network** exists. Recurrent Neural Networks(RNN) allow to process sequential input with help of sequential propagation of the signal through a series of cells. It is interesting that RNN shares weights across all cells. A scheme of the cell is present in Figure6 and its stacking in Figure7.

A problem with RNN[5] is that gradients that flow through the model are vanishing from cell to cell. The illustration of vanishing gradients is given in Figure 8. To solve this problem, new versions of RNN called Long Short Term Memory(LSTM)

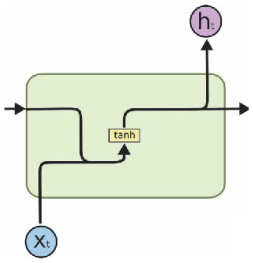

Figure 6: Scheme of a single RNN cell

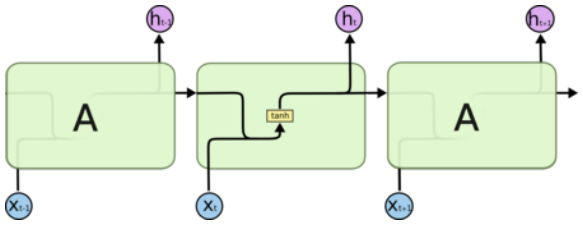

Figure 7: Recurrent neural network

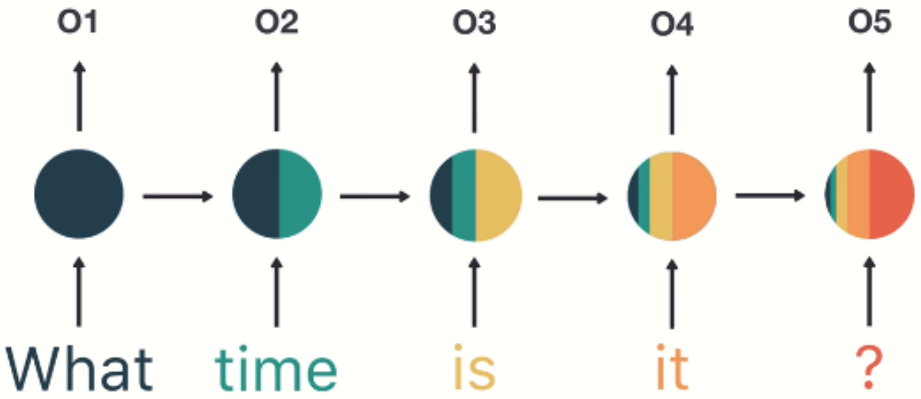

Figure 8: An illustration of the vanishing gradient problem - Image by deeplearning.ai

and Gated Recurrent Unit(GRU) were introduced. Their cells use a Gate mechanism (see Figure 9) to regulate the flow of information to keep or discard it in the loop. The main difference between these two is that GRU has 2 gates, while LSTM has 3 and GRU does not possess an internal memory.

Sequence models, such as RNNs, can be used for a variety of tasks like texts classification or sequence-to-sequence (id. seq2seq) modelling. Seq2seq models input and output text sequences, They are applied in neural machine translations, Part-of-speech tagging, next sentence prediction, and so on. In the era of RNNs, typical

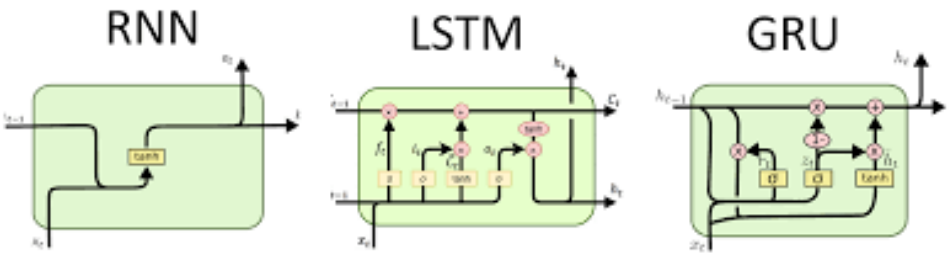

Figure 9: A comparison of the RNN cell to the LSTM and GRU cells

Seq2seq pipelines were utilizing LSTMs. LSTMs could capture longer sequences with better numeric stability. Despite that, processing large paragraphs was still an issue due to an excessively large amount of information needed to be fed into *embeddings*[6]. Managing large sequences of words requires tracking the relation between words.

Definition 3.1.1. According to[7], a token is a sequence of characters grouped to form a semantic unit. Tokens represent the words of a text. They can be the same as their word or a processed version, for example, a root of the word.

Definition 3.1.2. An embedding is a finite vector representation of a word of the token. Roughly, this vector carries the general meaning of this word "embedded" from a space of words vocabulary into some d-dimensional space.

3.2 "Attention is all you need"

Attention as a mechanism which is a part of Transformer was originally introduced in 2017 by Vaswani et al.[8] and it started a new era in NLP. This novel design presents encoder-decoder architecture comprising a brand new attention mechanism.

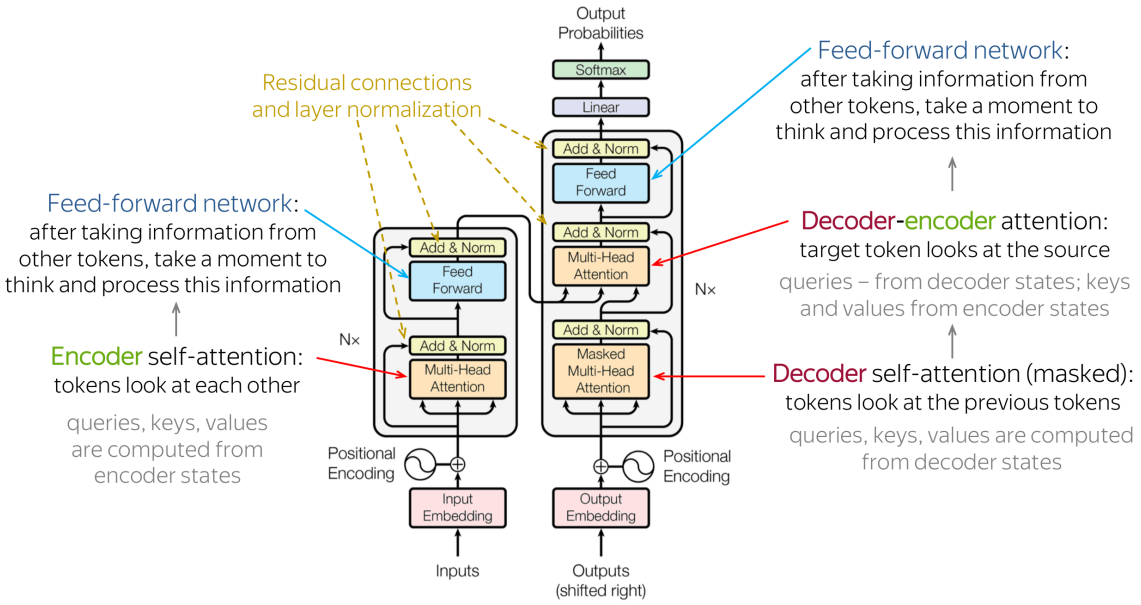

Figure 10: The Transformer - model architecture

For a given sequence of words, attention measures the "tightness" of relations between different words. Attention works like a retrieval system where it compares words against each other just like it works in information search engines. A concrete explanation can be observed in the original paper on Attention[8] or a detailed illustrated article about Transformers in [9].

A few words explanation of how Transformers operate is given in Figure 10.

Transformers that are based on this mechanism appear to be able to work on multiple NLP tasks including NMT. Since transformers require much higher computational power, they are commonly used with already pretrained weights.

3.3 Transformers: T5, BERT, GPT

Since attention was introduced several multimodal (see 3.3.1) models were developed and shared wtih the public.

Definition 3.3.1. Multimodality is an ability of a model to solve different tasks.

Bidirectional Encoder Representation for Transformer[10] (BERT) that was trained on Masked Language Modelling (see definition 3.3.2) task, able to give quality embeddings. It was made in such a paradigm that it can be adapted to a specific task by just redesigning the output and thus eliminating the need to retrain it from scratch for further use.

Definition 3.3.2. Masked language modelling(MLM) is a training approach to teach a language model to predict a masked word in a given text sequence. For example, we have the sentence *"London is the capital of Great Britain."* and we mask the word "capital" by replacing it with a special token ¡MASK¿, so that the goal of the model is to predict this word.

General Pre-trained Transformer[11] (GPT) is also of great interest. It is trained on a next sentence prediction task (see definition 3.3.3) and it also follows a fashion of being redistributed with pre-trained weights to be used for downstream tasks. Specifically, GPT can be applied in such NLP problems as chatbots, speech writing, news reports and so on.

Definition 3.3.3. Next sentence prediction task(NSP) is another training approach that teaches the model to predict tokens of the next sentence given an input one. NSP allows the model to "understand" dependencies between sentences.

T5[12] is a model that was a result of a training text-to-text transformer on multiple supervised and unsupervised tasks. It is capable to adjust the output by adding prefixes to input sequence, for example, to translate: "translate French to Italian:"[French text], to summarize: "summarize:"[long sequence].

We will review transformer-based approaches in this work since they show state-of-the-art performance on various tasks and benchmarks. Transformers are also known for solving two tasks named entity recognition (NER) and Relation Extraction (RE). Some papers refer to both of them as a composite task named the same Relation Extraction, Joint Relation Extraction, or End-to-End Relation Extraction. We will explore what is a RE task in the next chapter.

4. Entity and Relation Extraction

Relation Extraction is an NLP task and is used in a pipeline to build knowledge graphs. Having entity-relation-entity triples extracted from a raw text we can construct a graph interconnecting different entities with various relations.

Let us define what named entity recognition and relation extraction are how to collect a dataset for RE and NER and what are the possible approaches for these tasks.

4.1 Named Entity Recognition and Relation Extraction

When talking about NER and RE, we must first consider defining a span.

Definition 4.1.1. Span is a part of a text comprising one or more words and representing single instance.

Span is commonly referred to when there is a NLP task that considers marking a subsequence of words in a sentence to assign it to some category or to detect that words represent certain nature. One of the most frequent spans is entity. It usually represent well-known places, organizations and persons. In more specific domains it can be definitions like "sulfuric acid", "Naegleria fowleri" or "diode".

Definition 4.1.2. Named entity recognition is a NLP task to detect named spans in a natural text. Entity recognition, a more general task, can focus on finding subjects and objects in text.

BIO encoding	Michel B-PER	Jordan I-PER	would O	choose O	Bush B-PER
BILOU encoding	Michel B-PER	Jordan L-PER	would O	choose O	Bush U-PER

Figure 11: Example of BIO/BILOU tagging

A widely used approach to classify named entities in texts is BIO/BILOU tagging. **BIO/BILOU** sequence tagging aims to teach a seq2seq model to output an aligned tag sequence that is the same size as the input. Each tag represents a position inside an entity: *B - Beginning token, I - Inside token, O - Outside and entity (L - Last token, U - Unit-length entity).* These tags can be also combined with entity types like organization, person or place. For example, "Alber Einstein" will be tagged as *B-PER, I-PER.*

Entities in the text can have relations between them that we want to extract to use in a downstream task of KG construction. Relations can be found using a) a rule-based method which relies on dependency parsing or keywords or b) using supervised methods.

Definition 4.1.3. Relation Extraction is the task of finding and classifying relations for pairs of entities in a given input text. E.g. given a sequence "HSE University was found in 1992" it outputs relation "found_in" for entities spans "HSE University" and "1992".

Sometimes, it is important to track if entities have coreferences (or mentions)

in the text. The task to find these coreferences is called **coreference resolution**. Coreference resolution allows to find entities in text that were mentioned differently from their original name and use them to find more potential relations.

Now, it is clear that to construct a knowledge graph a subtask of entity and relation extraction should be solved. To train RE systems we need training data. Let us explore what RE datasets exist.

4.2 Training data

The relation extraction task collected several datasets around it which are used as public benchmarks for various models.

4.2.1 Existing public datasets

TACRED

TACRED[13] is a large-scale human-annotated crowdsourced dataset for relation extraction comprising 106,264 samples built over newswire and web text from the corpus used in the yearly TAC Knowledge Base Population (TAC KBP) challenges. As, in TAC KBP challenge TACRED has 42 relation types, accounting one "no_relation" type. Stanford proposes to use TAC RED as a benchmark and/or train data for KBP systems.

DocRED

DocRED[14] (Document-Level Relation Extraction Dataset) consists of data from Wikipedia and Wikidata. All samples are divided in two parts: 1) 5,053 human-annotated Wikipedia docs with 132,375 entities, 56,354 relations, and coreference information and 2) 101,873 distantly supervised documents. DocRED is especially interesting for measuring the ability of the models to extract entities and relations across different documents.

ADE

Adverse Drug Effect is a dataset designed to extract adverse effects from consuming drugs described in medical reports Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. It is divided into 3 subsets: first for text classification and the other two for relation classification and extraction. One interesting fact is that this dataset does not contain a tail entity meaning the RE tasks would be unary relation extraction.

NYT-H

Zhu et al.[15] provided a distantly supervised dataset called NYT-H built on top of NYT-10 from 2010. Authors propose an alignment strategy to automatically annotate relations and entities using a knowledge base such as Freebase. NYT-H is significantly larger then its predecessor and can serve as a benchmark for RE models.

RURED

A collective of researchers from Moscow RANEPA[16] built a dataset for RE from Russian texts. The authors chose Lenta.ru news articles, specifically the Economic section, as a data source. This dataset contains 500 annotated texts and 5000 relations with a baseline F1-score of 0.85 for NER and 0.78 for RE.

4.2.2 Problems of RE datasets

Collecting quality RE dataset is not an ordinary task. Human annotation is costly and relatively slow if we compare it to distantly supervised annotations. On the other hand, the latter is noisier and less accurate. Still, it is possible to pretrain the model on a distant-supervision dataset.

The other problem lies in relation classification task which is a subtask for relation extraction. Relation types are not uniformly distributed in natural texts. Consider Figure 12 below. We see that the relation types distribution is very heavy-tailed. It will be a major problem in the performance of relation classification.

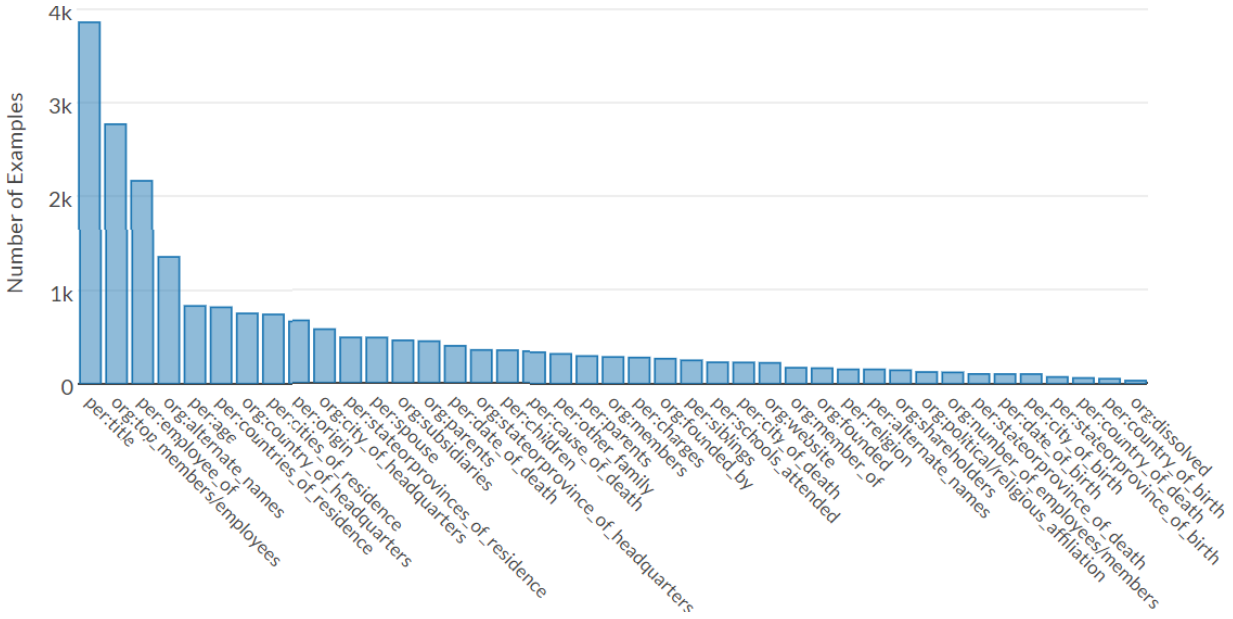

Figure 12: Heavy tailed distribution of the TACRED relation types - Image from [13]

4.3 Approaches to relation extraction

Now that we know what datasets exist, we can think of how we can solve RE task. As it was well described by Gordeev et al.[16], relation extraction approaches divide in two:

- supervised learning-based methods
- distant supervision-based methods

While first is a base for building distant supervision datasets, it is not capable of finding new entities that are not present in a knowledge base. Thus, to be able to au-

tomatically extract relations from the unstructured text we need to follow supervise methods.

In CS520[3] V.K. Chaudhri describes 3 different approaches to building RE systems:

- sequence labelling
- language modelling
- rule-based

Sequence labelling focuses on training token classification algorithms, such as CRFs, but it takes time for severe feature engineering. The second approach is to adapt a language model, particularly by using new boundary tokens to emphasize spans in text. This model will learn to output the boundary token and thus predict an entity or a relation. The last one is the rule-based approach and its name describes its logic: to extract a span we may use dictionaries, regular expressions or use other extraction tools.

We know that the language models, specifically Transformers are good learners for a variety of NLP tasks. Latest works focus more on using this architecture. We will review how to employ Transformers to extract relations from texts in the next Section.

5. Current study on relation extraction

In this chapter we will provide an overview of current study of the relation extraction task.

Sequence tagging methods

One of straightforward approaches to relation extraction is an employment of the sequence tagging technique. Different[17, 18] works on joint entity and relation extraction use the BIO or BILOU labels(see Figure 11).

A problem of sequence tagging models is that they cannot accurately tag multiple relations in the sentence due to limitations of tagging. They also fail to deal with overlapping entities. Say we have a sentence *"Sberbank First is a premium service for wealthy citizens."*. Entities *"Sberbank"* and *"Sberbank First"* have common tokens and thus we cannot share **B** and **I** tags between entities. If we set *"Sberbank - B"* and *First - I*, we will omit *"Sberbank"* entity. On the other hand *"Sberbank - B"* and *"First - O"* leaves *"Sberbank First"* entity untagged.

Span-based methods

Since BIO/BILOU models suffer from overlapping entities there should be another approach to consider folded spans. Span-based approaches perform a so-called "exhaustive search over all spans". This gives significant improvements on more tasks like coreference resolution. Some models using a span-based approach have

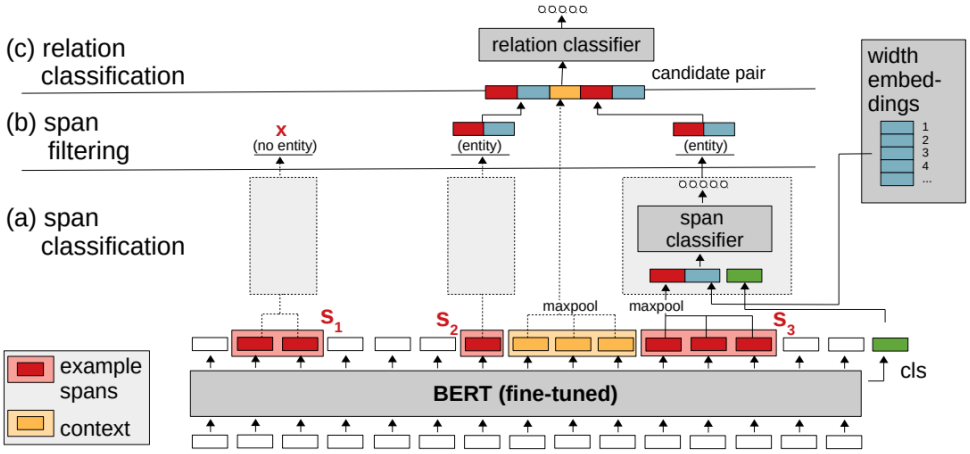

Figure 13: Architecture of SpERT - Image from paper

been proposed in [19, 20]. They use a Bidirectional version of LSTMs from ELMo embeddings.

SpERT[21] that was introduced by Elberts et al. utilizes a pre-trained BERT model. It uses two classifiers: one for span classification and the other to classify relations of span pairs. On Figure 13 we can see SpERT's architecture. It is divided into 3 stages. In the first stage, all spans in a sequence are classified into entity types, which are filtered in the next stage where *(*no_entity) class is omitted. During the last stage, all pairs of entities leftover are classified for a relation. As seen in the figure, the span classifier uses two more inputs besides span embedding: span width embedding and sentence context embedding. Similarly, the relation classifier takes 3 concatenated inputs: 2 embeddings of a span pair and a context embedding derived from tokens between these two spans.

Authors also mention that a negative sampling that makes SpERT learn better is also of crucial importance. Let us observe an example from the paper: having a sen-

tence "In 1913, Olympic legend $[Jesse\,Owens]_{People}$ was born in $[Oakville, Alabama]_{Location}$." negative samples such as "Owens" or "born in" can be taken. Experiments on datasets CoNLL04, SciERC, and ADE showed the growth of F1-score by up to 2.6% by the time of publication.

Graph-based methods

Spans can be dealt differently compared to [21]. Several latest studies utilize graph-based approaches to perform relation extraction. Some of them[22, 23] focus on resolving mention-mention coreferences. However these approaches do not take mention-pronoun pairs into account, thus they do not take some beneficial information into account.

In [24] Xue et al. emphasized two major problems in the relation extraction task: 1) performing coreference resolution not only with direct mentions of a particular entity, but also with pronoun mentions 2) capturing relations on the document level.

The authors decided to focus on explicit modelling of mention-pronoun pairs for RE across multiple sentences. As it was in the aforementioned papers this modelling was performed with graph-based methods. Generally, CorefDRE works on finding mentions and pronouns in text and merging them if they are representing the same entity. Thus, it is possible to find relations that refer to certain entities indirectly through their pronoun mentions.

Let us take a look at Figure14. Steps to extract relations are as follows: 1) Feed the document in the encoder (e.g. BERT), from where a heterogeneous graph is constructed with pronoun and mention nodes 2) Perform coreference resolution over

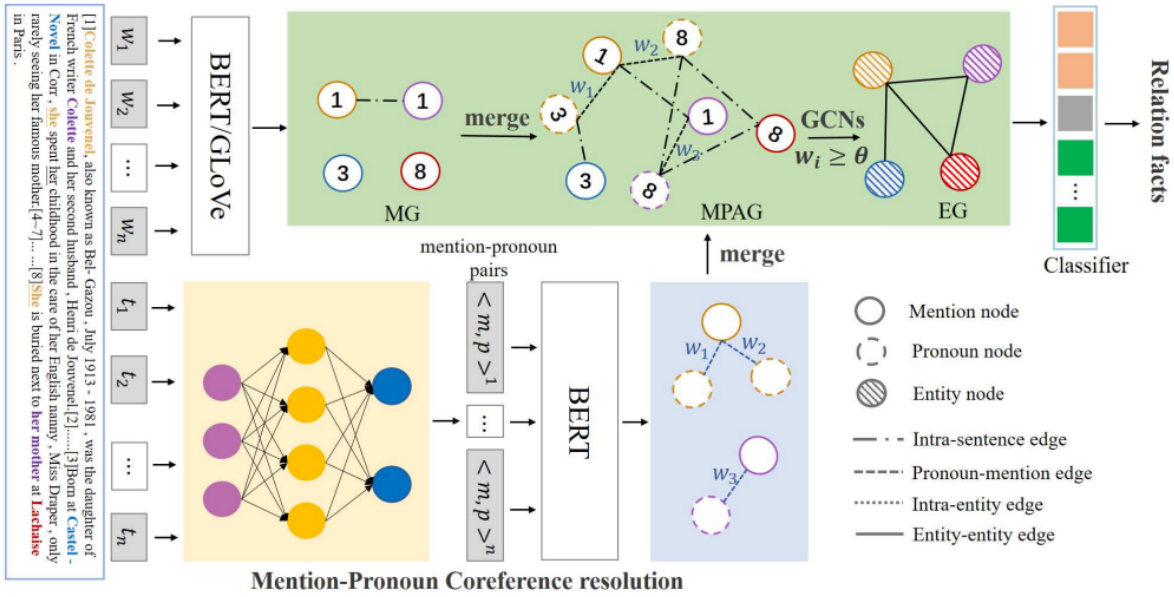

Figure 14: Architecture of CorefDRE - Image from paper

mention-pronoun pairs which calculates affinity scores. 3) Apply these scores on hetero-graph to perform merging of mention pairs 4) The result is an entity graph, where edges representing relations can be classified. Experiments showed that this method gets a 60.82 F1-score on the DocRED dataset which is relatively close to the other state-of-the-art results.

Seq2seq methods

Seq2seq is a vast class of methods of different nature that rely on reducing the RE task to outputting text sequences to learn relations. Papers like [25, 26] propose a decoding mechanism to output entities more than once. There are also attempts to reframe seq2seq Transformers models to different NLP tasks for Entity Linking

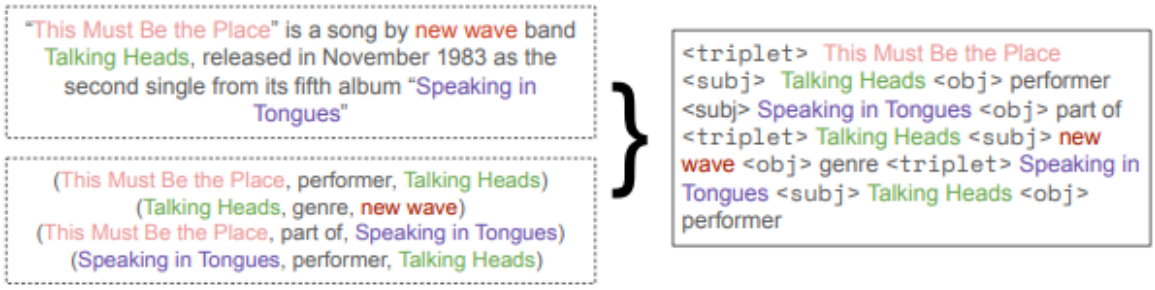

Figure 15: Example of linearization - Image from paper

or Semantic Role Labelling [27, 28]. The relation extraction task is not an exception here.

Huguet et al.[29] proposed a remarkable approach. They transform a task of relation extraction to sequence generation with help of marker tokens. They invented a **linearization** technique that builds a target sequence based on triples in an input text. To generate the target sequence one needs to find all subjects, then each subject should be emphasized by $< subj >$ and $< /subj >$ tokens, followed by emphasized relation and object spans. In other words, the final sequence will be roughly an adjacency list of subject spans. To understand better we should refer to Figure 15.

Given that input of a model is a raw text and output is a linearized triples as on Figure 15 authors suggest training such seq2seq transformer model as BART. BART is an autoregressive sequence generator meaning that each output token is dependent on previous tokens. According to the paper, equitation for conditional probability is as below:

$$p_{BART}(y|x) = \prod_{i=1}^{len(y)} p_{BART}(y_i|y_0, ..., y_{i-1}, x) \tag{5.1}$$

BART model is tuned on this task using Cross-Entropy Loss as in Machine Translation. The resulted model is called REBEL.

Authors also provided a tool for creating distantly-supervised datasets called CROCODILE[30]. It uses RoBERTa model[31] and data from Wikidata and Wikipedia to construct a dataset. Huguet et al. emphasized the benefits of pre-training REBEL on such a dataset and showed it gives significant improvement to the performance compared to REBEL with no pretraining.

One more important fact is that REBEL is trained on relatively low resources: a single NVIDIA 3090 GPU with 64GB of RAM and Intel® Core™ i9-10900KF CPU as it is mentioned in the paper. At the moment of writing this work, REBEL holds 3 state-of-the-art results on NYT, DocRED, and CoNLL04 benchmarks and 2 third places on ADE and RE-TACRED.

5.1 Selecting approach

Our KG construction pipeline will utilize the approach described in the REBEL paper. It has two three advantages: REBEL is simple yet effective, BART has a multilingual version that can be pre-trained with CROCODILE generated dataset and the entire pipeline can be trained on relatively small resources.

6. Implementation of Knowledge Graph construction

Preparing training data and a source text

As in the REBEL paper we build a pretraining dataset constructed from Wikidata and Wikipedia dumps. Huguet et al. made a relatively convenient tool for extracting triples in a single json-file of a dataset. We follow instructions provided in a CROCODILE repository[30] with a small changes considering that our environment for building a dataset is Google Colab and Kaggle. REBEL-RU dataset is publicly available at
huggingface.co/datasets/InfroLab/REBEL-RU.

DocRED and RURED datasets are delivered in formats different from REBEL-RU. We use their respective authors' preprocessing scripts provided in project repositories. Note that RURED is a domain-specific dataset constructed on the economical news texts. It does not represent a general domain. It has a relatively low number of samples, especially compared to datasets like TACRED. This is actually a problem that Gordeed et al. described in their paper: RE task lacks quality Russian datasets.

For a phase of KG inference, we propose to use clear texts that do not contain markup tags, markdown styles and so on. The model itself is able to extract relations from large paragraphs. These relations will form the final KG.

Model architecture

The base model for REBEL is the $BART_{large}$ and it was made by Meta AI team and trained on English texts. We need a model capable of dealing with Russian texts

too. Meta AI also offers a multilingual version of the $BART_{large}$ that was pretrained on a same task with a corpus having 50 languages. It is called $mBART_{large}$-50. We preserve the Cross-Entropy Loss for our task as it was proposed by Huguet et al.

Training the model and Experimental setup

Our model will be trained on the hyperparameters provided in parameters file from REBEL repository. Note that a pretraining stage that fits REBEL-RU dataset has a different configuration. To train the model we use a ML server with 8xNVIDIA Tesla v100 GPUs, 512 GB RAM and 96 Intel CPUs. Despite its power, the server, we need to mention that it is operated as communal enterprise resource with several other users.

Model evaluation

As for the evaluation step, we propose to measure F1-score on following datasets: DocRED and RURED. The last one would be a main indicator of performance for extracting relations for Russian language. Since RURED is built from economical news, we would also be able to evaluate the performance of the model on a domain-specific data. Additionaly, we suggest to compare the evaluation metrics with baselines defined in RURED paper.

Demonstration of knowledge graph construction

Since $mBART_{large}$-50 is a multilingual model we expect that the it will be able to generate accurate triples for both: English and Russian. The model is capa-

ble of outputting linearized sequence of relations and then transform it to a triples list to construct the KG then. For demonstration purposes we will use 4 articles from Wikipedia: *Artificial Neural Network, Physician, Democracy, Higher School of Economics.*

We will provide evaluation results, 4 pairs of graph demo images and a source code in a Git repository github.com/InfroLab/hse-thesis.

7. Future work

In a future work, one can explore a way to address a problem of coreference resolution in a more optimal way than it was done in [24]. It may be performed with an enhancement of a linearization technique of REBEL or any explicit way that will model coreferences. This can be done by further employment of special tokens to highlight mentions and pronouns in text and tie them to the main entity span.

Papers described in the Section 5 do not explicitly model logical implications or, in other words, relations that are a logical consequence of what is written in text. This can be approached from the KG perspective, where given one set of relations another relation is added. From point of view of the RE task, it may require new datasets, that annotate "hidden" relations.

There can be also advancements in multilingual relation extraction that may suggest a new training approach to learning the same relations across different languages efficiently. Some facts can be described in other languages and they can be used to populate knowledge graphs constructed in the others. This can be done by either exchanging information between KGs constructed from different languages or by inventing a new transfer learning approach.

Finally, a bigger initiative can be launched to construct a new RE dataset that will solve the problem mentioned in subsection 4.3. A new dataset creation tool may be created by combining machine translation and word alignment techniques to translate datasets to other languages.

8. Conclusion

This thesis reviewed aspects of the Knowledge Graphs from comparison to relational databases and enterprise application to construction using modern NLP methods. We contributed an overview of a current study of an upstream relation extraction task and provided a dataset for RE built from Russian Wikidata and Wikipedia. As we know, this dataset is known the be the largest distant-supervised dataset for RE. In the end, we proposed a pipeline to train and apply the model for building KGs from texts. It can be utilized for downstream production tasks. Although, it is worth mentioning that we were unable to propose any relatively large dataset to test the pipeline on a general domain.

An important thing to emphasize is that further work on this project requires high-performance equipment and a significant amount of time. Enhancing the results of RE requires a richer general domain dataset with a more accurate annotation.

Bibliography

[1] B. I. C. Education, "What is a knowledge graph?."

[2] V. Ryen, A. Soylu, and D. Roman, "Building semantic knowledge graphs from (semi-)structured data: A review," Apr 2022.

[3] *CS520: 2021 Knowledge Graphs Seminar Session 10.* Vinay K Chaudhri, Apr 2021.

[4] N. Chittar, V. K. Chaudhri, and M. Genesereth, "An introduction to knowledge graphs," May 2021.

[5] V. Lendave, "Lstm vs gru in recurrent neural network: A comparative study," Dec 2021.

[6] K. Loginova, "Medium: Attention in nlp."

[7] C. D. Manning and H. Schütze, *Introduction to Information Retrevial - Section Tokenization.*

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017.

[9] J. Alammar, "The illustrated transformer."

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.

[11] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[12] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *CoRR*, vol. abs/1910.10683, 2019.

[13] Y. Zhang, V. Zhong, D. Chen, G. Angeli, and C. D. Manning, "Position-aware attention and supervised data improve slot filling," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pp. 35–45, 2017.

[14] Y. Yao, D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou, and M. Sun, "DocRED: A large-scale document-level relation extraction dataset," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 764–777, Association for Computational Linguistics, July 2019.

[15] T. Zhu, H. Wang, J. Yu, X. Zhou, W. Chen, W. Zhang, and M. Zhang, "Towards accurate and consistent evaluation: A dataset for distantly-supervised relation extraction," *CoRR*, vol. abs/2010.16275, 2020.

[16] D. I. Gordeev, A. A. Davletov, A. I. Rey, G. R. Akzhigitova, and G. A. Geymbukh.

[17] G. Bekoulis, J. Deleu, T. Demeester, and C. Develder, "Joint entity recognition and relation extraction as a multi-head selection problem," *CoRR*, vol. abs/1804.07847, 2018.

[18] D. Q. Nguyen and K. Verspoor, "End-to-end neural relation extraction using deep biaffine attention," *CoRR*, vol. abs/1812.11275, 2018.

[19] Y. Luan, L. He, M. Ostendorf, and H. Hajishirzi, "Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction," *CoRR*, vol. abs/1808.09602, 2018.

[20] K. Dixit and Y. Al-Onaizan, "Span-level model for relation extraction," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 5308–5314, Association for Computational Linguistics, July 2019.

[21] M. Eberts and A. Ulges, "Span-based joint entity and relation extraction with transformer pre-training," *CoRR*, vol. abs/1909.07755, 2019.

[22] S. Zeng, R. Xu, B. Chang, and L. Li, "Double graph based reasoning for document-level relation extraction," *CoRR*, vol. abs/2009.13752, 2020.

[23] D. Ye, Y. Lin, J. Du, Z. Liu, M. Sun, and Z. Liu, "Coreferential reasoning learning for language representation," *CoRR*, vol. abs/2004.06870, 2020.

[24] Z. Xue, R. Li, Q. Dai, and Z. Jiang, "Corefdre: Document-level relation extraction with coreference resolution," 2022.

[25] X. Zeng, D. Zeng, S. He, K. Liu, and J. Zhao, "Extracting relational facts by an end-to-end neural model with copy mechanism," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Melbourne, Australia), pp. 506–514, Association for Computational Linguistics, July 2018.

[26] T. Nayak and H. T. Ng, "Effective modeling of encoder-decoder architecture for joint entity and relation extraction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 8528–8535, Apr. 2020.

[27] N. D. Cao, G. Izacard, S. Riedel, and F. Petroni, "Autoregressive entity retrieval," in *International Conference on Learning Representations*, 2021.

[28] N. D. Cao, G. Izacard, S. Riedel, and F. Petroni, "Autoregressive entity retrieval," in *International Conference on Learning Representations*, 2021.

[29] P.-L. Huguet Cabot and R. Navigli, "REBEL: Relation extraction by end-to-end language generation," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, (Punta Cana, Dominican Republic), pp. 2370–2381, Association for Computational Linguistics, Nov. 2021.

[30] Babelscape, "Babelscape/crocodile: Crocodile is a dataset extraction tool for relation extraction using wikipedia and wikidata presented in rebel (emnlp 2021).."

[31] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019.

[32] D. Ye, Y. Lin, and M. Sun, "Pack together: Entity and relation extraction with levitated marker," *CoRR*, vol. abs/2109.06067, 2021.

[33] E. Dezhic, "Understanding knowledge graphs," May 2018.

[34] "Attlasian: Knowledge graph," Jun 2021.

[35] C.-W. Lin, Y. Shao, J. Zhang, and U. Yun, "Enhanced sequence labeling based on latent variable conditional random fields," *Neurocomputing*, vol. 403, 05 2020.