Algorithms and methods for automated construction of knowledge graphs based on text sources

Victor Filippov[1*], *Natalya* Ayusheeva[1], and *Maria* Kusheeva[2]

[1] East Siberia State University of Technology and Management (ESSUTM), Applied computer science, statistics and data analysis Department, 670013 Ulan-Ude, Russian Federation
[2] Tomsk Polytechnic University (TPU), Information Technology Department, 634050 Tomsk, Russian Federation

Abstract. In this article, we present our path towards building knowledge graphs automatically from Russian texts. We explore various methodologies and libraries to extract triples, which are the fundamental building blocks of knowledge graphs. Our approach involves the use of libraries for analyzing morphological characteristics of words, such as PyMorphy and Yandex Mystem, to construct triples. We also utilize the NLP library spaCy to analyze text and build triples based on semantic relationships recognized by the library. However, we found that in some cases, we could not extract relationships from the text, leading us to use word2vec to define relationships. Unfortunately, the results obtained from word2vec were unsatisfactory and could not be used as relationships. We also encountered the problem of building triples from text due to the use of pronouns. To address this issue, we explored the use of coreference resolution libraries, but unfortunately, there are no working libraries available for the Russian language at this time. Our results highlight both positive and negative outcomes of applying these methodologies and libraries, providing insights into the challenges and opportunities of building knowledge graphs automatically from Russian texts.

1 Introduction

Knowledge graphs are among the most demanded structures for organizing and representing complex information and knowledge. In the age of information technology, processing a large amount of information presented in natural language has become easier because of the emergence of large computing power, which makes it possible to automate the processes of knowledge extraction and construction of knowledge graphs. The automatic construction of knowledge graphs is a complex and challenging task. At the moment there are no open tools for building knowledge graphs in automatic mode on the basis of Russian-language texts, since it is textual data, being the most popular means of information storage and distribution, reflect the wealth of potential knowledge. In this regard, the task of automatic construction of knowledge graphs is urgent. To solve this problem, it is necessary to investigate the

* Corresponding author: vitya.filippow@yandex.ru

possibility of using various text processing tools in order to identify effective tools for automatic construction of knowledge graphs.

2 Knowledge graphs

The term "knowledge graph" was introduced as early as 1972 in a discussion on how to create modular learning systems for courses. In the late 1980s, the University of Groningen (The Netherlands) and the University of Twente (The Netherlands) jointly started a project called Knowledge graphs, focusing on the design of semantic networks for which a limited set of relations was specified. This approach facilitated the application of algebras on the graph.

Early knowledge graphs were mostly topic-based. In 1985, Wordnet was founded, capturing semantic relations between words and their meanings. In 1998, Andrew Edmonds of Science in Finance Ltd in the UK created a system called ThinkBase that offered reasoning based on fuzzy logic in a graphical context. In 2005, Mark Virk founded Geonames to capture the relations between different place names and localities and related objects. In 2007, DBpedia and Freebase were founded as graph-based knowledge repositories for representing general-purpose knowledge. DBpedia focused exclusively on the data extracted from Wikipedia [1]. Freebase incorporated data from Wikipedia as well as a number of publicly available datasets [2]. None of these projects called itself a "knowledge graph", but essentially developed and described related concepts. In 2012, Google introduced their knowledge graph based on DBpedia and Freebase, among other sources. Later, they included data extracted from indexed Web pages, including the CIA World Factbook, Wikidata, and Wikipedia. The entity types and relations used in this knowledge graph were further organized using terms from schema.org. The Google Knowledge Graph has become a successful addition to Google text-based search, and its popularity on the Internet has led to a wider use of the term.

The definition of a "knowledge graph" in research articles and practice-analytic articles ranges from specific technical definitions to more comprehensive general definitions [3]. In this paper, a comprehensive definition is adopted, which considers a knowledge graph as a data graph designed to accumulate and communicate knowledge about the real world, with nodes representing objects of interest and edges representing relations between these objects. A data graph corresponds to a graph-based data model, which can be an orientated graph with labelled edges, a property graph, etc. [4]. At the same time, researchers in their work define a knowledge graph as a type of semantic network [5]. Both semantic networks and knowledge graphs are graph models that describe entities and their relations. The key difference is the level of detail and complexity. Knowledge graphs often contain more detailed information and are used to solve more complex problems. They can be seen as an extension or a more detailed version of semantic networks.

– The concept of knowledge graph is a powerful tool that solves complex problems in various domains. It provides a structured representation of knowledge and facilitates the following tasks:

– information retrieval; knowledge graphs organize a huge amount of data, which makes the search for specific information more efficient. They support search engines and recommender systems [6];

– semantic search; knowledge graphs improve search engines by understanding the context and semantics of queries, which provides more accurate and context-aware search results [6];

– recommendation systems; knowledge graphs support personalized recommendations in e-commerce, content delivery systems, and social networks by identifying relations between users and objects in the graph [6];

– data integration; knowledge graphs integrate data from different sources, providing a unified view of information. This is critical in fields such as healthcare and finance [7];

– ambiguity resolution; knowledge graphs help in resolving entity ambiguities, which is an important task in data cleaning and resolving entity ambiguities in various applications [7];
– question-answering systems; knowledge graphs provide complex question-answering systems by linking relevant nodes of information and relations, helping in natural language understanding [8];
– machine learning; knowledge graphs enhance machine learning models by providing structured background knowledge and making their operation explainable through graph engineering [9];
– explainable artificial intelligence; knowledge graphs improve the interpretability of machine learning models, making their predictions and decisions more transparent and reliable [9];
– knowledge representation; knowledge graphs provide a basis for representing complex knowledge, helping in expert systems, chatbots, and knowledge systems [10];
– network analysis; knowledge graphs facilitate network analysis to identify hidden patterns, detect anomalies, and optimize systems such as transportation and social networks [11].

Knowledge graph representation is an integral task in the realization of various applications. The choice of representation format depends on the specific use case as well as the scalability and expressiveness requirements of the knowledge graph. The machine-readable format used in knowledge graphs is usually based on semantic technologies such as RDF and OWL. These technologies provide a standardized way to represent knowledge about objects and their relations using a graph structure, allowing knowledge to be easily shared between different systems and applications.

RDF (Resource Description Framework) is a World Wide Web Consortium (W3C) standard originally developed as a data model for metadata. It has come to be used as a common method for describing and exchanging data represented as a graph, making it a powerful tool for organizing structured information within a knowledge graph. Data in RDF is represented in the form of triplets - triples of the form "subject-predicate-object" [12]. Each part of an RDF triplet is individually addressed via unique URIs [12]. This representation allows semantic data to be unambiguously queried and analyzed. To work with RDF in the context of knowledge graphs, the SPARQL language (SPARQL Protocol and RDF Query Language) is used. It is a semantic query language for databases that is able to retrieve and manipulate data stored in RDF format, specifying patterns and conditions that the data should fulfill [12]. SPARQL queries can be used to extract, update and manipulate RDF data, making it a key technology for semantic applications.

OWL (Web Ontology Language) is a semantic web language used to create ontologies, which are formal representations of knowledge that are readable by both humans and machines. OWL is designed to provide a way to express complex relations and dependencies between concepts in a structured and standardized way, allowing machines to understand and reason about the meaning of data on the web. OWL is based on RDF and uses a set of vocabularies and syntaxes to define classes, properties and individuals and the relations between them. OWL allows users to define their own vocabularies and use existing ones, making it a flexible and extensible language.

Automated construction of knowledge graphs is a complex and challenging task, for which the following sequence of basic activities can be identified.

Text preprocessing. As in all text analysis methodologies, it is necessary to perform preprocessing of input data. In the framework of the knowledge graph construction task, it can be the selection of separate sentences from the text, resolution of co-referentiality, replacement of punctuation marks with their semantic analog. For example, in the sentence "Solotcha (-) is a winding, shallow river", the dash sign can be replaced by the lexeme "is". Since many algorithms and methods are trained directly on lexemes or terms, and punctuation marks are weeded out of the training set, replacing punctuation marks can positively affect

the result of knowledge graph construction, since lexemes or terms carry more information than punctuation marks.
1. Entity extraction; identifying entities (e.g., people, organization, location) in text that can be interpreted as objects and subjects of the knowledge graph.
2. Relation extraction; identifying relations between entities mentioned in the text.
3. Triples formation; creating triplets of the form "subject-predicate-object" representing the extracted entities and their relations. These triples will be the basis of the knowledge base.
4. Graph construction; constructing a knowledge graph based on the knowledge base.
Once the knowledge graph is constructed, several additional procedures can be identified to improve the quality of the knowledge graph.
1. Knowledge base enrichment; enriching the knowledge base by adding additional information such as attributes of entities.
2. Visualizing the graph for better understanding.
3. Regular updating: populating new knowledge and updating old knowledge.

3 Morphological vectors of word features for building triples

In some works, methods based on the analysis of morphological characteristics of words have been used to extract entities from texts for the purpose of automatic construction of semantic networks. The use of morphological word vectors to identify triplets in the context of knowledge graph construction can be justified based on several key points:
1) rich linguistic information; morphological vectors collect detailed linguistic information about words, including their prefixes, suffixes, and other morphological properties. This information can be valuable when trying to extract structured knowledge from unstructured text;
2) similarity and relatedness of words; morphological vectors can help determine semantic similarity and relatedness between words. This is important when trying to establish relations between entities in a knowledge graph. For example, words with similar morphological features may indicate a certain level of semantic similarity;
3) morphological patterns; certain morphological patterns may indicate certain relations or associations between words. For example, in Russian, words with the same root often share a common semantic domain. Recognizing these patterns can help in distinguishing triplets;
4) reliability across languages; morphological analysis is applicable across languages. Although some languages may have complex morphological structures, the approach can be adapted according to the linguistic characteristics of the target language;
5) complementary approach; morphological vectors can complement other triplet extraction methods. By combining morphological information with the results of syntactic and semantic analysis, a more complete understanding of the relations within the text can be achieved;
6) adaptability to the subject domain; the morphological vector approach is independent of the subject domain. It can be applied to a wide range of text sources and domains, making it versatile for knowledge graph construction in different contexts;
7) interpretable results: morphological vectors often produce interpretable results. Analysts and researchers can understand the basis on which triplets are extracted, which can be important for quality control and validation;
8) resource efficient; compared to more complex natural language processing methods, morphological analysis can be computationally more efficient. This makes it a practical choice for processing large amounts of textual data.

The above advantages have conditioned the use of morphological features of words to construct text-based knowledge graphs. Using the morphological parsing library Pymorphy2 equipped with the Russian language corpus gives good results when processing texts of more general topics. At the same time, the Russian corpus of this library was not sufficiently

adapted for processing words from specialized subject areas. The application of the Yandex Mystem library demonstrated high efficiency in processing various texts of different subject areas (Fig. 1).

```
Word: Эзофагогастродуоденоскопия ; Normal form: эзофагогастродуоденоскопия ; Features: S,жен,неод=им,ед
Word: ЭГДС ; Normal form: эгдс ; Features: S,ед,муж,неод=(пр|вин|дат|род|твор|им)
Word: проводится ; Normal form: проводиться ; Features: V,несов,нп=непрош,ед,изъяв,3-л
Word: всем ; Normal form: весь ; Features: APRO=(дат,мн|пр,ед,муж|пр,ед,сред|твор,ед,муж|твор,ед,сред)
Word: пациентам ; Normal form: пациент ; Features: S,муж,од=дат,мн
Word: при ; Normal form: при ; Features: PR=
Word: отсутствии ; Normal form: отсутствие ; Features: S,сред,неод=пр,ед
```

Fig. 1. Examples of morphological features of words extracted using the Yandex Mystem library

Thus, to identify morphological features of words in the future you can use the Yandex Mystem library.

4 Using NLP libraries to construct triples

To extract entities expressed by collocations and to extract relations, the results of syntactic analysis of text sentences are often used. In order to determine the cohesion of sentence members, the NLP capabilities of the spaCy library have been considered (Fig. 2).

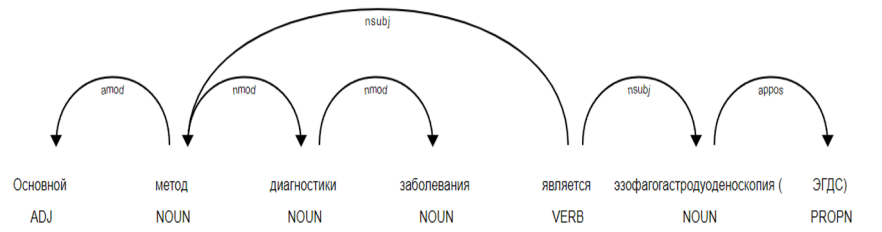

Fig. 2. Proposal analysis results using the spaCy library

In addition to its powerful relation extraction function, the spaCy library has morphological analysis capabilities similar to the Yandex Mystem library. The library's methods identify parts of speech and tag them with labels from the following set:
- ADJ: adjective or ordinal numeral;
- ADP: preposition/postposition;
- ADV: adverb;
- AUX: auxiliary verb;
- CONJ: conjunction;
- CCONJ: co-ordinate conjunction;
- DET: indicative pronoun;
- INTJ: interjection;
- NOUN: noun;
- NUM: numeral;
- CASE: particle;
- PRON: pronoun;
- PROPN: proper name;
- PUNCT: punctuation mark;

- SCONJ: subordinating conjunction;
- SYM: symbol;
- VERB: verb;
- X: other;

The set of relations defined with the spaCy library includes NSUBJ - the subject of the sentence (the actor or proto-agent of the sentence), OBJ - the object affected, NMOD - the dependent noun, AMOD - the property expressed by the adjective, APPOS - the extension of the noun or its abbreviation. In total, the library allows defining forty different relations.

The above features allowed a deeper exploration of the linguistic characteristics of words and paved the way for the development of a system of rules for the extraction of compound noun phrases. The rules are strongly depends on the implementation of the syntactic parsing of the spaCy library. Examples of rules are:

1. If the token is a noun and it is related to another noun by an NMOD relation, then identify the two words as a single term expressed by the collocation. If the linked noun is also linked to a noun phrase, then consider the three words as a single term. Produce linking nouns into a single term until the last noun no longer refers to a noun phrase. In this way we can extract complex terms of the form "method of diagnosing a disease", "examining a condition", etc.

2. If the token is a noun and refers to an adjective, with the semantic relation parameter AMOD, and the adjective has the morphological feature Case=Ins, then identify the two words as a complex term. This rule is aimed at extracting word combinations such as "fatty tissue", "hypertension", etc.

Further analysis of the semantic relations in the sentence identified using the spaCy library revealed a pattern that if two terms refer to the same word, it is most likely a relation. Thus, in Figure 2, the words "method" and "esophagogastroduodenoscopy" refer to the word "is", indicating the presence of a functional relation. Taking into account the terms extracted with the use of the system of rules and the identified relation, we manage to compose a triplet: "Esophagogastroduodenoscopy - is - a method of diagnosing diseases" (Fig. 3).

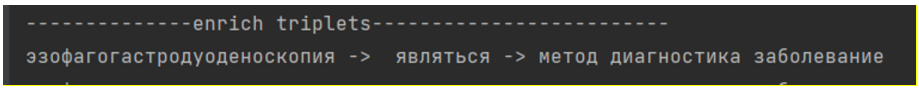

Fig. 3. Example of a selected triplet

The combined capabilities of describing rules for extracting terms and defining rules enable the extraction of complex knowledge from unstructured content. However, in order to realize a complete mechanism on knowledge graph construction, a number of the following problems need to be solved:

1) linking triplets from different sentences into a graph; in the context of the task of linking triplets from different sentences into a knowledge graph, coreference resolution becomes a key aspect. This is due to the fact that pronouns serve as a link between the extracted entity from the previous sentence and the new knowledge from the current sentence. Unless the system is able to accurately identify which word replaces a pronoun, it is impossible to build a complete knowledge graph;

2) identifying the relation between subject and object; it is a non-trivial task, especially when the relation in the sentence is expressed implicitly or alternative forms are used. For example, instead of the word "is" there may be a hyphen symbol. This ambiguity requires the search for tools and the development of algorithms capable of detecting and interpreting different forms of relation expression, even if they are not explicitly stated in the text;

3) dealing with conjunctions; currently, the rule system is based on the assumption that two entities referring to the same word can have a relation with each other. However, this logic does not always take into account the context associated with conjunctions and complex

allied sentences. In order to interpret relations more accurately, further research and development of methods capable of analyzing and accounting for conjunctions in text is needed, thus expanding the system's ability to detect and understand complex grammatical constructions. For example, the system currently identifies "esophagogastroduodenoscopy (EGDS) and biopsy" as a triplet "esophagogastroduodenoscopy biopsy - to be - method" when analyzing the sentence "The main methods of diagnosing the disease are esophagogastroduodenoscopy (EGDS) and biopsy". The conjunction "and" is expected to produce two triplets instead of one: "biopsy - to be - method" and "esophagogastroduodenoscopy - to be - method".

5 Using word2vec to define relations

Since not all relations between words in the text are explicitly expressed, there is a need to develop methods for their additional detection. For this purpose, it is necessary to use external sources of knowledge. The word2vec (bag of words) model can be used as an external knowledge source. This model is capable of identifying syntactically close words between two given words based on the average distance between them.

However, it should be noted that the results obtained using word2vec are characterized by a certain degree of abstraction and their interpretation as explicit relations requires additional analysis. For example, for the pair of words "method" and "diagnosis", the word2vec model proposes to use the relation "methodology", which may be somewhat unexpected and uninformative (Fig. 4).

```
метод_NOUN  -  ('методика_NOUN', 0.7778502702713013)  -  диагностика_NOUN
методика_NOUN 0.7778502702713013
диагностический_ADJ 0.7060539722442627
система_NOUN 0.6558462977409363
анализ_NOUN 0.6551244258880615
исследование_NOUN 0.6467984914779663
диагностирование_NOUN 0.6429771184921265
подход_NOUN 0.6403197050094604
детекция_NOUN 0.6214499473571777
технология_NOUN 0.6164605617523193
селективный_ADJ 0.6156601905822754
```

Fig. 4. Using word2vec to define object and subject relations

Moreover, further analysis shows that the next nine words suggested by the word2vec method as potential relations for this pair are also incorrect and not suitable to be used as relations. The most appropriate word to represent the relation between subject and object is often a verb. Verbs indicate actions or relations between entities in a sentence and can be more informative in the context of relation extraction.

Although Word2vec is a powerful tool for determining semantic similarities between words, it is limited in its ability to accurately interpret relations between words. Sometimes it may suggest synonyms or associations, but not necessarily exact relations. Also, word2vec results can be highly dependent on context and domain of knowledge. Therefore, it is important to consider the text context and adapt the method to a specific domain if necessary. Further research may be needed to identify subject-object relations in sentences more accurately and reliably. This is an important issue in the field of knowledge extraction from textual data and its solution may require innovative approaches and techniques

6 Coreference resolution

To ensure effective communication between triplets in different segments of the text, it becomes necessary to identify cases where pronouns are used to denote antecedents. The process of co-referentiality resolution, i.e., determining what the pronouns refer to, facilitates the establishment of links between text fragments that may share a common entity or use pronouns to denote it. The study found a natural language processing library developed by Stanford University. This library is available in two versions: for the Python language (stanza) and for the Java language (stanfordnlp).

The Python version of the library (stanza) has specialized models customized for Russian. However, it should be noted that this version lacks the functionality of co-referentiality resolution, which is a critical feature when analyzing texts in which pronouns and their corresponding antecedents are used.

On the other hand, the Java version of the library (stanfordnlp) has coreference resolution functionality, which allows us to determine what a pronoun refers to in a text. However, a challenge arises here due to the lack of trained models for Russian.

It is important to emphasize that the search for alternative efficient tools for co-referentiality resolution has not yielded conclusive results so far. Therefore, more research is required in this area. Effective coreference resolution remains a key aspect for creating more accurate and comprehensive knowledge graphs from textual data.

7 Conclusion

In this paper, various methods and tools have been investigated to build knowledge graphs from textual data. The main method of knowledge extraction is semantic analysis using the spaCy library, which is capable of identifying semantic relations between words in a sentence, as well as determining the morphological vector of a word. Further working with textual data and creating knowledge graphs, can be useful in various domains where automatic construction of knowledge graphs based on textual sources is required.

Although Word2vec is a powerful tool for determining semantic similarities between words, it is limited in its ability to accurately interpret relations between words.

An important conclusion is the need for further research and development of methods for extracting semantic relations from textual data. It is also important to continue the search for effective tools for coreference resolution, especially in the context of working with the Russian language.

References

1. D. Abián, et al. Wikidata and DBpedia: a comparative study, Springer International Publishing, 142-154 (2018)
2. T. T. Pellissier, et al., From freebase to wikidata: The great migration, Proceedings of the 25th international conference on world wide web, 1419-1428 (2016)
3. S. Tiwari, F. N. Al-Aswadi, D. Gaurav, Recent trends in knowledge graphs: theory and practice, Soft Computing, vol. **25**, 8337-8355. (2021)
4. A. Hogan et al., Knowledge graphs, ACM Computing Surveys (Csur), **54**, 1 – 37 (2021)
5. C. Gutierrez, J. F. Sequeda. Knowledge graphs, Communications of the ACM, vol. **64**, 96-104 (2021)

6. C. Peng, F. Xia, M. Naseriparsa, et al. Knowledge Graphs: Opportunities and Challenges, Artifcial Intelligence Review, vol. **56**, 13071–13102 (2023).
7. M Kejriwal, Knowledge Graphs: A Practical Review of the Research Landscape, Information, vol. **13**, 17 (2022)
8. S. Srivastava, et al. Complex Question Answering on knowledge graphs using machine translation and multi-task learning, Proceedings of the 16th conference of the European chapter of the association for computational linguistics: Main Volume, 3428-3439 (2021)
9. I. Tiddi, S. Schlobach, Knowledge graphs as tools for explainable machine learning: A survey, Artificial Intelligence, vol. **302**, 21 (2022)
10. V. Chaudhri, et al, Knowledge graphs: introduction, history and perspectives, AI Magazine, vol. **43**, 17-29 (2022)
11. Y. Chen, et al, An overview of knowledge graph reasoning: key technologies and applications, Journal of Sensor and Actuator Networks, vol. **11**, 26 (2022)
12. W. Ali, M. Saleem, B. Yao, A. Hogan, A.N. Ngomo, A survey of RDF stores & SPARQL engines for querying knowledge graphs, The VLDB Journal, **vol. 3**, 1-26 (2021)
13. I. Melnyk, P. Dognin, P. Das, Knowledge Graph Generation From Text, Association for Computational Linguistics, 1610–1622 (2022)
14. StableLM., Official Web-Site, access mode: https://stablelm.ru/
15. A. Candel, J. McKinney, P. Singer, et al., h2oGPT: Democratizing Large Language Models, arXiv preprint, **22** (2023)
16. B. Peng, E. Alcaide, Q. Anthony, et al., RWKV: Reinventing RNNs for the Transformer Era, Findings of the Association for Computational Linguistics: EMNLP 2023, 14048–14077 (2023)
17. RWKV Language Model, Official Web-Site, access mode: https://www.rwkv.com/