Bull. Nov. Comp. Center, Comp. Science, 49 (2025), 65-80
@ 2025 NCC Publisher

From text to knowledge: Entity linking of scientific terms to Wikipedia

Daniil Kuzovlev, Tatiana Batura

Abstract. This paper addresses the critical task of entity linking for scientific terms in Russian texts to Wikipedia, a process vital for transforming unstructured text into structured knowledge. We introduce a novel algorithm and conduct a comparative analysis between RuBERT-tiny2 and spaCy, evaluating their performance across varying context window sizes and numbers of links. Our findings indicate that RuBERT-tiny2 excels with larger context windows, leveraging its deep semantic understanding for superior disambiguation, though its performance degrades beyond 100 tokens due to noise. Conversely, the spaCy-based approach demonstrates greater robustness in limited-context scenarios. This highlights a trade-off: while complex models like RuBERT-tiny2 are highly context-dependent, simpler models remain competitive when contextual information is sparse. Error analysis reveals three primary failure modes: search errors (absence of correct entities), ranking errors (suboptimal semantic scoring), and annotation errors (ambiguities in ground truth). The study underscores the direct impact of knowledge base quality on system performance and suggests implementing semantic similarity thresholds to mitigate overconfident false links. We conclude that future advancements in entity linking necessitate not only improved algorithms but also enhancements in candidate retrieval, query formulation, ranking strategies, and robust handling of ambiguous annotations, advocating for adaptive thresholding, dynamic context selection, and domain-specific knowledge integration.

Keywords: information extraction, entity linking, Wikipedia, scientific text processing, concept normalization

Introduction

Scientific texts have always been and remain the primary source of new knowledge. However, with the increasing volume of published materials, there arises a problem of effective information extraction and structuring. One of the key tasks in natural language processing is entity linking (EL), which involves matching mentioned concepts in texts with corresponding entries in knowledge bases [1]. This article is devoted to the study of this task important for scientific texts, where terms refer to words or phrases used within a specific domain to precisely denote particular concepts, phenomena, or objects; entities refer to the elements (entries) in the knowledge base. Solving this task opens up new possibilities for creating intelligent systems capable of automatically analyzing scientific publications, extracting key concepts, and linking them to existing structured knowledge.

Entity linking of terms to knowledge base entities is a complex task, the main difficulties of which are associated with term ambiguity, differences in terminology across disciplines, and the necessity of considering the context in which terms are used within the text. Current EL methods actively employ contextual augmentation to improve linking accuracy. The authors of [2] propose an approach based on Large Language Models (LLMs) that generate additional context for entity mentions, helping to resolve ambiguities. The method includes dynamic context expansion of the query and semantic ranking of candidates, which is especially useful for short or uninformative mentions. An important advantage is its adaptability to different knowledge domains, making the approach universally applicable to specialized knowledge bases.

The paper [3] presents a scalable EL method that works for 100 languages, including rare and low-resource ones. The authors use bidirectional transformers (namely, mBERT) and train the model on data with automatically generated annotations (so-called "weakly labeled data"), employing knowledge distillation and cross-lingual transfer techniques. A key feature of the approach is the efficient use of multilingual embeddings, which enables high accuracy even for languages with limited training data.

The aforementioned works require vast computational resources, making efficient and lightweight EL methods relevant under limited computing conditions. The authors of [4] propose a system called ReLiK, which combines fast candidate retrieval based on indexed embeddings with accurate ranking using lightweight neural architectures. This approach demonstrates competitive accuracy at significantly lower computational costs compared to LLMs, making it promising for academic research.

The paper [5] describes a method based on Dense Entity Retrieval (DER). Instead of traditional approaches using sparse representations (e.g., TF-IDF or BM25), the authors employ neural embeddings to encode both textual mentions and entity descriptions from the knowledge base. This approach does not require training on labeled data specific to a domain, as it relies on general semantic representations, making it especially useful for rare or new terms. Such methods are promising for the Russian language, as they help overcome the problem of limited annotated corpora by relying on multilingual embeddings (e.g., from mBERT).

Overall, it can be observed that the majority of research in this area is focused on the English language, with only a limited number of studies dedicated specifically to Russian [6, 7]. However, research on Russian is highly relevant due to the need for the tools that account for its rich morphology and the scarcity of annotated data. This paper proposes a new algorithm for linking terms from Russian scientific texts to entities in Wikipedia. Experiments were conducted on texts from four knowledge domains: information technology, medicine, psychology, and linguistics.

1. Related work

In the traditional approach, entity linking is carried out in two stages: candidate generation and candidate ranking.

Candidate Generation. Candidate generation in entity linking is the process of selecting potential entities from the knowledge base that may correspond to a given term in the text. This step reduces the search space, enabling more efficient ranking in the next stage.

There are several approaches to candidate generation. One of them is the dictionary-based approach [8–10]. A dictionary of names implies a mapping from a set of keys to a set of possible candidates (values). For example, the term "*field*" is mapped to a set of candidates including entities such as "*algebraic field*", "*magnetic field*", "*agricultural field*", and so on.

Another approach to candidate generation is the word form expansion technique. In [11, 12], the term is enriched with contextual information.

Statistical information on term occurrences across various texts can also be used for candidate generation. Based on this statistical data, a prior probability of the entity is calculated. Prior probability provides an empirical estimate of the likelihood that a term refers to a specific entity. For example, in [13], the authors propose an approach based on the use of prior distributions.

Methods based on name dictionaries or prior probability computation require training data. In such cases, pre-filled name dictionaries or pre-collected statistical information about entities is used. These methods are not suitable for knowledge domains where sufficient training data is unavailable.

In this paper, the Wikipedia search engine is used for candidate generation, allowing reliance on the existing knowledge base with the same name. A similar approach is used in [14]: dictionaries based on Wikipedia are applied to search for entities with similar names (accounting for abbreviations, synonyms, and spelling variations), semantic expansion of candidates using context (leveraging keywords and other mentions within the document), and resorting to search engines if the first two methods do not yield a sufficient set of candidates.

However, other search engines can also be used. For example, in [12, 15, 16], the Google search engine is employed for candidate generation, while [17] uses Bing. Subsequently, a filtering step is applied to the search results — only pages from Wikipedia are selected.

This paper proposes an approach for working with unlabeled data. A similar approach is used in [18], where the authors present a method for training an EL model without annotated data. The paper employs surface matching between a term and the entity name in the knowledge base. In surface matching, for each term, an entity is searched for in the knowledge base whose name contains all the words of the term. This differs from the approach used in this paper, where candidate entities are selected from the top n candidates retrieved via a Wikipedia API search query.

Candidate Ranking. Candidate ranking is the process of ordering a list of potential entity candidates by the degree of their relevance to a given term in the text. This stage follows candidate generation and is used to select the most relevant entity, taking into account the context. Candidate ranking involves measuring the similarity between the term and entities in the knowledge base. As a measure of similarity, dot product or cosine similarity between vector representations of the term and the entity is typically used.

Recently, attention mechanisms have been increasingly applied [13, 19–21]. In [13], all words in the context and candidate entities are mapped into a vector space using pre-trained word embeddings. The ranking of candidate entities is performed in two stages: first, the local context of the term is analyzed, and then the results are refined by considering all entities in the document. The model uses an attention mechanism to select important words and takes into account relations between entities to improve prediction accuracy, helping to avoid local errors and ensure consistent labeling.

A similar two-stage approach is used in [21]. The authors propose an iterative approach to candidate ranking, where entities are linked sequentially, using the terms already identified to refine the context. Two methods for entity selection are considered: selection by highest confidence (confidence-order) and selection in the order they appear in the text (natural-order). Experiments show that the first method performs better, as it allows for the accumulation of global context, leading to more accurate entity linking.

In [20], candidate ranking is performed using a cross-encoder, which combines the term context and the description of each entity into a single input for a BERT model. The model then evaluates the probability of each entity matching the given term, and the final ranking is based on the output logits. This method significantly improves accuracy compared to a bi-encoder, although it requires more computational resources.

The authors of [19] propose replacing the traditional two-stage entity linking process with an end-to-end method, in which both terms and entities are encoded into the same vector space. Unlike previous approaches, the model completely eliminates the use of precompiled alias tables. Candidate retrieval is performed by encoding the term into the vector space, followed by an approximate nearest neighbor search among pre-encoded entities based on cosine similarity. An end-to-end method based on the BERT model is also explored in [22].

In this paper, both a Convolutional Neural Network (CNN) and BERT are employed to generate vector representations of terms from Russian-language texts and Wikipedia entities. Although transformer-based models, such as BERT, have become widely used for entity linking [20, 22], approaches based on convolutional networks have also been applied successfully, as shown in previous studies [23, 24]. The decision to incorporate a CNN in the current work was based on several factors. First, convolutional neural networks are known to be effective at capturing local dependencies in data, which is essential for understanding context. Second, CNNs have been shown to perform well even on relatively small datasets—a particularly valuable property in the tasks where labeled data is limited, as is the case in this study. Third, while transformer-based models may offer higher accuracy, they require significantly more computational resources and training time. In this regard, CNNs are considered less resource-intensive and faster to train, due to their ability to process data in parallel.

2. Formal definition of the entity linking task

The entity linking task is defined as the process of mapping the mentions of entities appearing in a text document to their corresponding canonical representations in a knowledge base. Given a document D, which contains a set of entity mentions denoted as $M = \{m_1, m_2, \dots, m_n\}$, and a knowledge base $E = \{e_1, e_2, \dots, e_k\}$, such as Wikipedia, where each entity $e_j \in E$ is uniquely identifiable and associated with a description (including synonyms, semantic types, and other attributes). The task is to determine, for each entity mention m_i , the most appropriate entity e_j , i.e., to find a mapping [1]

$$f: M \to E \cup \{NIL\},$$

where $f(m_i) = e_j$ if the mention m_i refers to the entity e_j , and $f(m_i) = NIL$ if no suitable entity exists in the knowledge base. The NIL value is used to indicate that a

[1] It is important to note that multiple correct mappings may exist for a single mention. In cases where a mention is ambiguously associated with several entities in the knowledge base, a mapping is considered correct if it links the mention to any of the valid corresponding entities.

mention cannot be linked to any known entity. This formulation encapsulates the central challenge of disambiguating entity references in text with respect to a given knowledge base.

It should be noted that, as in [25], we focus only on the entity linking task, since the named entity recognition (NER) step was performed beforehand and is not addressed in our study. Also, in [25] each entity is represented by its title and description – though additional fields exist in the database, only these two are used in the model. Embeddings are generated based on the description field. This is conceptually similar to our approach, where we use the "snippet" field of each Wikipedia entity as the basis for embedding generation. This field serves a function analogous to the description used in [25], providing a concise textual summary of each entity.

Despite these similarities, there are several key differences between the two approaches. The authors of [25] rely on a locally stored knowledge base, whereas we use Wikipedia as our primary source of entity information, accessed dynamically via its API. This choice is beneficial because Wikipedia is continuously updated and maintained by a global community, ensuring that the knowledge base remains current. In contrast, local knowledge bases tend to become outdated unless they are regularly updated, which can be resource-intensive. Moreover, the size and scope of a local knowledge base are inherently limited. The knowledge base used in [25] includes only 13,125 entities, whereas Wikipedia provides access to millions of articles across a wide range of domains.

3. Data description

For conducting experiments, we compiled a dataset consisting of abstracts from scientific articles available in open access. The dataset includes texts in Russian from four scientific fields: information technology, medicine, psychology, and linguistics. The average text length is 216 words, and the total number of terms is 367.

Before inputting the data into the entity linking algorithm, preliminary processing was performed, which involved four stages:

- term annotation,
- term extraction,
- morphological cluster identification,
- deduplication.

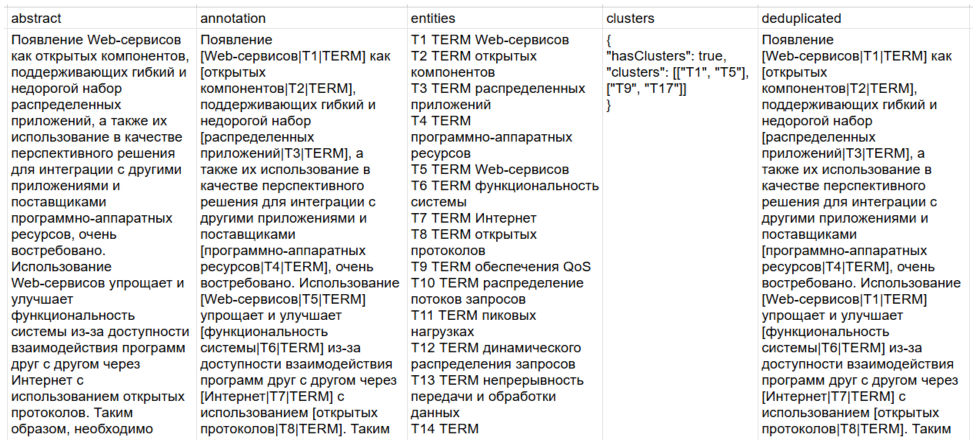
1) *Term annotation*. In the scientific abstracts, two types of entities were annotated: TERM and VALUE. The TERM entities refer to words or phrases used within a specific domain to denote precisely particular concepts, phenomena, or objects. For example, in the field of information technology, the terms include names of methods, architectures, models, programming languages, etc., while in medicine, they include the names of diseases, symptoms, drugs, diagnostic procedures, and so on. Abbreviations are also considered terms. The VALUE entities are numerical values combined with additional information (context or unit of measurement), such as quantitative or qualitative indicators used to describe specific data that can be measured or evaluated.

Each entity is assigned a unique identifier. Entities are annotated in the [*Term* | *Identifier* | *Label*] format. The label TERM indicates that the selected word sequence is a term; the label VALUE is used for numerical values.

The original texts were initially annotated using the *gpt-4o-mini* model. This was followed by manual correction, with two annotators working on each text. To avoid inconsistent annotations, a detailed annotation guideline with descriptions and examples was developed in advance. In the final stage of annotation, a moderator resolved any remaining ambiguous cases according to this guideline. Inter-annotator agreement was measured using the standard statistical measure — Cohen's kappa. For the prepared dataset, an average kappa value of 0.73 was obtained, indicating high annotation quality [26].

2) *Term extraction.* For convenience in further processing, terms (entities of type TERM) are separated from the annotated text and extracted into a dedicated block. In the entity linking algorithms, only TERM-type entities are considered; VALUE-type entities are not taken into account.

3) *Morphological cluster identification.* A term may appear multiple times in a text or in different word forms (e.g., different cases, singular and plural forms). For example (see Fig.1), the term "Web-сервисов" appears twice in a text. After the term detection stage, two separate entities would be identified: "*Web-сервисов | T1 | TERM*" and "*Web-сервисов | T5 | TERM*", each with its own unique identifier. During the morphological clustering stage, identifiers of the same term and its various word forms are grouped into a single cluster.



abstract	annotation	entities	clusters	deduplicated

Появление Web-сервисов как открытых компонентов, поддерживающих гибкий и недорогой набор распределенных приложений, а также их использование в качестве перспективного решения для интеграции с другими приложениями и поставщиками программно-аппаратных ресурсов, очень востребовано. Использование Web-сервисов упрощает и улучшает функциональность системы из-за доступности взаимодействия программ друг с другом через Интернет с использованием открытых протоколов. Таким образом, необходимо	Появление [Web-сервисов|T1|TERM] как [открытых компонентов|T2|TERM], поддерживающих гибкий и недорогой набор [распределенных приложений|T3|TERM], а также их использование в качестве перспективного решения для интеграции с другими приложениями и поставщиками [программно-аппаратных ресурсов|T4|TERM], очень востребовано. Использование [Web-сервисов|T5|TERM] упрощает и улучшает [функциональность системы|T6|TERM] из-за доступности взаимодействия программ друг с другом через [Интернет|T7|TERM] с использованием [открытых протоколов|T8|TERM]. Таким	T1 TERM Web-сервисов
		T2 TERM открытых компонентов
		T3 TERM распределенных приложений
		T4 TERM программно-аппаратных ресурсов
		T5 TERM Web-сервисов
		T6 TERM функциональность системы
		T7 TERM Интернет
		T8 TERM открытых протоколов
		T9 TERM обеспечения QoS
		T10 TERM распределение потоков запросов
		T11 TERM пиковых нагрузках
		T12 TERM динамического распределения запросов
		T13 TERM непрерывность передачи и обработки данных
		T14 TERM
			{
			"hasClusters": true,
			"clusters": [["T1", "T5"], ["T9", "T17"]]
			}
				Появление [Web-сервисов|T1|TERM] как [открытых компонентов|T2|TERM], поддерживающих гибкий и недорогой набор [распределенных приложений|T3|TERM], а также их использование в качестве перспективного решения для интеграции с другими приложениями и поставщиками [программно-аппаратных ресурсов|T4|TERM], очень востребовано. Использование [Web-сервисов|T1|TERM] упрощает и улучшает [функциональность системы|T6|TERM] из-за доступности взаимодействия программ друг с другом через [Интернет|T7|TERM] с использованием [открытых протоколов|T8|TERM]. Таким

Figure 1. Data processing pipeline

4) *Deduplication.* In the annotated text, duplicate occurrences and all word forms of the same term are assigned the same identifier. The unique identifier is selected as the first one from the morphological cluster formed in the previous step. Deduplication allows for unambiguous identification of each term.

4. Entity linking algorithm

The proposed entity linking algorithm consists of three main stages:

- Searching for links in Wikipedia;
- Calculating semantic similarity;
- Ranking the links found.

Searching for a specified number of Wikipedia links by term. Interaction with Wikipedia is carried out through its API. The input data for the search are the term and the number of links. For each found entity, the API returns a short description of the link's content in the snippet field. This field is used to calculate semantic similarity in the next step of the algorithm. An example response is shown in Appendix A.

Calculating semantic similarity between the short description of the link's content and the term. The semantic similarity between the target term and the candidate links found in the first step is computed based on the descriptions from the snippet field of the Wikipedia API response using cosine distance. For this purpose, both the term itself and the text from the snippet field are represented in a vector form.

In this work, we examine two methods for obtaining vector representations. The first method uses the *ru_core_news_md* model, which is part of the spaCy library. The second method employs the BERT model [27]. Then, to determine semantic similarity, we calculate the cosine distance.

The **SpaCy-based method** consists of the following steps.

Tokenization. The input text is first tokenized using spaCy's default tokenizer (spacy.Tokenizer.v1)[2], which splits the text into individual tokens. These tokens are then stored in a Doc object – a structured container that holds all linguistic annotations[3].

Obtaining vector representations for each token in the text. The spaCy library does not provide the ability to train vector representations but uses pre-trained ones instead [4]. The *ru_core_news_md* model employs vector representations from the Navec library[5], which were obtained using the GloVe algorithm [28], trained on fiction literature, followed by a quantization of the vector representations[6].

Obtaining the vector representation of the text. The vector representation of the text is computed as the arithmetic mean of the vectors of the tokens contained in the text.

The **BERT-based method** uses the RuBERT-tiny2 model[7], which is a lightweight Russian-language version of the BERT model.

In the previous method, the vector representation of the text was obtained by averaging the vector representations of each token. Since BERT is a transformer-based model, this approach generates the text's vector representation through a more sophisticated process, leveraging the principles of self-attention.

Ranking the found links by semantic similarity. For each Wikipedia link found, the algorithm calculates the semantic similarity between the vector representation of the scientific term (along with its context) and the entity description from the snippet field for that link. Based on the obtained semantic similarity value, the Wikipedia links are ranked from the most semantically similar to the least similar. As a result of the ranking, the link with the highest semantic similarity value is selected and linked to the scientific term.

[2] https://spacy.io/api/tokenizer
[3] https://spacy.io/api/doc
[4] https://spacy.io/usage/embeddings-transformers#static-vectors
[5] https://github.com/natasha/navec
[6] https://natasha.github.io/navec/
[7] https://huggingface.co/cointegrated/rubert-tiny2

5. Experimental results

Experiments were conducted for different values of context window size and number of links. The main experimental results are presented in Table 1.

Table 1. Experimental results

SpaCy model				
Number of links	**Context window size**	**Precision, %**	**Recall, %**	**F1-score, %**
5	0	18.1818	22.7642	20.2166
	10	***19.4805***	***24.3902***	***21.6606***
	100	17.8571	22.3577	19.8556
	1000	17.8571	22.3577	19.8556
10	0	12.9870	16.2602	14.4404
	10	10.7143	13.4146	11.9134
	100	9.0909	11.3821	10.1083
	1000	8.1169	10.1626	9.0253
RuBERT-tiny2 model				
5	0	17.8571	22.3577	19.8556
	10	18.1818	22.7642	20.2166
	100	**21.7532**	**27.2358**	**24.1877**
	1000	19.1558	23.9837	21.2996
10	0	8.7662	10.9756	9.7473
	10	10.3896	13.0081	11.5523
	100	13.9610	17.4797	15.5235
	1000	11.3636	14.2276	12.6354

The best performance achieved by spaCy was 19.48% precision, 24.39% recall, and an F1-score of 21.66%. In comparison, the RuBERT-tiny2 model achieved higher precision of 21.75%, recall of 27.24%, and a comparable F1-score of 24.19%.

Across all experimental settings and parameter configurations, the performance of the RuBERT-tiny2 model is comparable to that of the spaCy-based approach, with the relative effectiveness of each method depending on the context window size. Specifically, RuBERT-tiny2 outperforms spaCy when larger context windows are used, which can be attributed to its ability to capture and utilize richer semantic information from extended textual context.

Interestingly, however, increasing the context window beyond a certain point leads to performance degradation. In particular, when the context size is expanded from 100 to 1000 tokens, the performance of RuBERT-tiny2 declines. This suggests that a context window of 100 tokens is sufficient for the model to make accurate linking decisions, and that additional context does not contribute meaningful signal. Instead, larger contexts may introduce extraneous or irrelevant information—commonly referred to as "noise"—which can interfere with the model's ability to correctly rank candidate entities.

In contrast, for smaller context windows, the spaCy-based model achieves better results, suggesting that its rule-based and syntactic features are more robust in data-scarce contexts, where deep contextual models like RuBERT-tiny2 cannot fully utilize their representational capacity. This indicates that the effectiveness of RuBERT-tiny2 is highly dependent on the availability of sufficient contextual information. When such information is limited, simpler models like spaCy may provide more reliable performance.

Our analysis has revealed a fundamental limitation tied to the completeness of the underlying knowledge base, which significantly impacts the overall system performance. In numerous instances, the initial retrieval stage, responsible for generating a list of candidate entities from Wikipedia, fails to include the correct entity altogether. When the ground truth link is absent from this candidate list, no subsequent step—regardless of its sophistication in disambiguation or ranking—can produce a correct linking result. This highlights that the quality and coverage of the knowledge base serve as an upper bound for the system's potential accuracy. Future work could explore strategies to augment the candidate generation process.

Following the candidate retrieval component, the ranking of these candidates contributes to the second most significant impact on the system's final accuracy. Errors at this stage lead directly to incorrect entity linking, even when the correct candidate was successfully retrieved. Improving the candidate ranking component is therefore a high-priority task for future development.

6. Discussion

Errors made by the entity linking system using Wikipedia as a knowledge base can be categorized into three main types:

- search errors, which occur when the correct entity is not retrieved during candidate generation;
- ranking errors, where the correct entity is among the candidates but is incorrectly ranked below others;

– annotation errors, which stem from inaccuracies or ambiguities in the ground truth data used for evaluation.

Search errors. This category includes cases where a term was linked to an incorrect entity due to peculiarities in the Wikipedia search algorithm. In such cases, the correct entity is missing from the search results. Two scenarios are possible: either the correct entity is absent from Wikipedia entirely, or the correct entity exists in Wikipedia but does not appear in the search results. In the first case, if the algorithm operates correctly, the term should remain unlinked. The second case is more interesting. A correct entity might not appear in the results for two reasons. The first reason is a peculiarity of the Wikipedia API search algorithm itself. For example, the entity may have been added to Wikipedia recently and has not yet been indexed (see the previous section on indexing). Therefore, such entities will not appear in search results. In this case, it is reasonable to consider the algorithm's behavior correct if the term remains unlinked. The second reason is an incorrectly formulated query. Errors of this type need to be analyzed separately to understand how queries can be adjusted to ensure that the correct entity appear in the search results. An alternative approach is to employ a different knowledge base that is better suited—or more complete—for the specific task at hand.

Ranking errors. This category includes cases where the correct entity is present in the search results, but during the ranking stage, it did not receive the highest semantic similarity score with the term. In such cases, it is necessary to examine the algorithm responsible for generating vector representations and computing semantic similarity. Potential remedies include fine-tuning the current model to produce more accurate embeddings or exploring the use of an alternative model altogether.

Annotation errors. This category includes cases where the annotator has linked a term incorrectly. Of particular interest are terms with ambiguous annotations. Sometimes even experts struggle to determine which entity corresponds to a given term. When multiple options are equally valid, it is reasonable to consider the algorithm's performance correct if it links the term to at least one of the valid candidate entities. For example, the term "*исследование*" ("*research*") could correctly refer to either https://ru.wikipedia.org/wiki/Ультразвуковое_исследование or https://ru.wikipedia.org/wiki/Клиническое_исследование. In some cases, the annotator may lack sufficient expertise in the subject area to judge whether a term has been linked correctly. For instance, the term "*гидроксилаза*" ("*hydroxylase*") might correspond to https://ru.wikipedia.org/wiki/21-гидроксилаза, but without expert medical knowledge, it can be difficult to evaluate the correctness of this link. To reduce annotation errors stemming from ambiguous or incorrect labeling, it is essential to provide annotators with detailed annotation guidelines and to ensure that the annotation task is performed by domain experts.

It should be noted that the completeness of the knowledge base affects the result quality. The algorithm links a term to an entity based on semantic similarity. Therefore, if the knowledge base does not contain an entity suitable for a given term, the algorithm will link the term to the entity with the highest semantic similarity. This approach may reduce the algorithm's accuracy, as the term will be linked to an incorrect entity. One possible solution is to introduce an empirically determined threshold for semantic similarity: if the semantic similarity between a term and an entity is below this threshold, the algorithm should not link them. Future research will include experiments to determine an optimal threshold value.

Conclusion

This paper studies the task of linking terms from scientific texts in Russian with entities of Wikipedia. We propose a new algorithm and present the results of experiments comparing RuBERT-tiny2 with spaCy across various context window sizes and number of links. The results show that RuBERT-tiny2 performs better with larger context windows, leveraging deep semantic features for more accurate disambiguation; however, performance degrades when the context exceeds 100 tokens, likely due to the introduction of irrelevant information or "noise." In contrast, the spaCy-based approach proves more robust in settings with limited context. The effectiveness of RuBERT-tiny2 is highly context-dependent, while simpler models remain competitive when contextual information is sparse, highlighting a trade-off between model complexity and contextual sufficiency in entity linking for Russian scientific texts.

Analysis of errors reveals three primary sources of failure: search errors, ranking errors, and annotation errors. Search errors arise when the correct entity is absent from candidate results, either due to gaps in the knowledge base or limitations in the Wikipedia API’s indexing and retrieval mechanisms. Ranking errors occur when the correct entity is retrieved but fails to be selected due to suboptimal semantic similarity scoring. Finally, annotation errors highlight inherent ambiguities in the ground truth data, especially in cases where multiple valid interpretations exist or where expert domain knowledge is required to make a correct judgment.

Importantly, the completeness and quality of the knowledge base—Wikipedia, in this case—directly impact system performance. When no suitable entity exists for a given term, the model may still produce an incorrect link based on the highest available (but insufficient) semantic similarity. To mitigate this, introducing a threshold for semantic similarity could prevent overconfident false links, allowing the system to abstain from linking when certainty is too low.

While modern contextual models like BERT significantly advance the state of entity linking, further improvements will require not only better algorithms but also enhancements in candidate retrieval, more robust query formulation, refined ranking strategies, and careful handling of ambiguous or missing annotations. Future work should focus on adaptive thresholding, dynamic context selection, and integration of domain-specific knowledge to further improve performance and reliability.

References

[1] Sevgili Ö., Shelmanov A., Arkhipov M., Panchenko A., Biemann C. Neural entity linking: A survey of models based on deep learning // Semantic Web Journal. – 2022. –No. 13(3). – P. 527–570.

[2] Vollmers D., Zahera H., Moussallem D., Ngomo A.-C.N. Contextual augmentation for entity linking using large language models // Proc. of the 31st International Conference on Computational Linguistics. ACL. – 2025. – P. 8535–8545.

[3] Botha J.A., Shan Z., Gillick D. Entity linking in 100 languages // Proc. of the 2020 Conference on Empirical Methods in Natural Language. ACL. – 2020. – P. 7833–7845.

[4] Orlando R., Cabot P.-L.H., Barba E., Navigli R. ReLiK: Retrieve and LinK, fast and accurate entity linking and relation extraction on an academic budget // Proc. of Findings of the Association for Computational Linguistics. ACL. – 2024. – P. 14114–14132.

[5] Wu L., Petroni F., Josifoski M., Riedel S., Zettlemoyer L. Scalable zero-shot entity linking with dense entity retrieval // Proc. of the 2020 Conference on Empirical Methods in Natural Language Processing. – 2020. – P. 6397–6407.

[6] Loukachevitch N., Artemova E., Batura T., Braslavski P., Ivanov V., Manandhar S., Pugachev A., Rozhkov I., Shelmanov A., Tutubalina E., Yandutov A. NEREL: a Russian information extraction dataset with rich annotation for nested entities, relations, and wikidata entity links // Language Resources and Evaluation. – 2024. – Vol. 58. – P. 547–583. https://doi.org/10.1007/s10579-023-09674-z.

[7] Mezentseva A.A., Bruches E.P., Batura T.V. Methods and techniques to automatic entity linking in Russian // Proceedings of ISP RAS. – 2022. – Vol. 34, No. 4. – P. 187–200. DOI 10.15514/ISPRAS-2022-34(4)-13, EDN TEYGLE (In Russian).

[8] Bunescu R., Pasca M. Using encyclopedic knowledge for named entity disambiguation // Proc. EACL. – 2006. – P. 9–16.

[9] Cucerzan S. Large-scale named entity disambiguation based on Wikipedia data // Proc. EMNLP-CoNLL. – 2007. – P. 708–716.

[10] Zhang W., Tan C.L., Sim Y.C., Su J. NUS-I2R: Learning a combined system for entity linking. – https://tac.nist.gov/publications/2010/participant.papers/NUSchime.proceedings.pdf.

[11] Zheng Z., Li F., Huang M., Zhu X. Learning to link entities with knowledge base // Proc. of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. – 2010. – P. 483–491.

[12] Han X., Zhao J. NLPR_KBP in TAC 2009 KBP Track: A Two-Stage Method to Entity Linking. – https://tac.nist.gov/publications/2009/participant.papers/NLPR_KBP.proceedings.pdf.

[13] Ganea O. E., Hofmann T. Deep joint entity disambiguation with local neural attention // Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing. – 2017. – P. 2619–2629.

[14] Fang Z., Cao Y., Li R., Zhang Z., Liu Y., Wang S. High quality candidate generation and sequential graph attention network for entity linking // Proc. of The Web Conference. – 2020. – P. 640–650.

[15] Monahan S., Lehmann J., Nyberg T., Plymale J., Jung A. Cross-lingual cross-document coreference with entity linking. – https://tac.nist.gov/publications/2011/participant.papers/lcc.proceedings.pdf.

[16] Dredze M., McNamee P., Rao D., Gerber A., Finin T. Entity disambiguation for knowledge base population // Proc. of the 23rd international conference on computational linguistics. – 2010. – P. 277–285.

[17] Cornolti M., Ferragina P., Ciaramita M., Schütze H., Rüd S. The SMAPH system for query entity recognition and disambiguation // Proc. of the First International Workshop on Entity Recognition & Disambiguation. – 2014. – P. 25–30.

[18] Le P., Titov I. Distant learning for entity linking with automatic noise detection // Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. ACL Anthology. – 2019. – P. 4081–4090.

[19] Logeswaran L., Chang M.W., Lee K., Toutanova K., Devlin J., Lee H. Zero-shot entity linking by reading entity descriptions // Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. ACL. – 2019. – P. 3449–3460.

[20] Wu L., Petroni F., Josifoski M., Riedel S., Zettlemoyer L. Scalable zero-shot entity linking with dense entity retrieval // Proc. of the 2020 Conference on Empirical Methods in Natural Language Processing. ACL. – 2020. – P. 6397–6407.

[21] Kolitsas N., Ganea O. E., Hofmann T. End-to-end neural entity linking // Proc. of the 22nd Conference on Computational Natural Language Learning. ACL. – 2018. – P. 519–529.

[22] Ayoola T., Tyagi Sh., Fisher J., Christodoulopoulos Ch., Pierleoni A. ReFinED: An efficient zero-shot-capable approach to end-to-end entity linking // Proc. of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track. ACL. – 2022. – P. 209–220.

[23] Francis-Landau M., Durrett G., Klein D. Capturing semantic similarity for entity linking with convolutional neural networks // Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. – 2016. – P. 1256–1261.

[24] Yang X. et al. Learning dynamic context augmentation for global entity linking // Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. – 2019. – P. 271–281.

[25] Zhang M., Van Der Goot R., Plank B. Entity Linking in the Job Market Domain. – 2024. – (Preprint / arXiv preprint; arXiv:2401.17979).

[26] McHugh M. L. Interrater reliability: the kappa statistic // Biochemia medica. – 2012. – No. 22(3). – P. 276–282.

[27] Devlin J. et al. BERT: Pre-training of deep bidirectional transformers for language understanding //Proc. of the 2019 conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. – 2019. – Vol. 1. – P. 4171–4186.

[28] Pennington J., Socher R., Manning C.D. Glove: Global vectors for word representation // Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing. – 2014. – P. 1532–1543.

Appendix A

Listing 1. Example response from Wikipedia API

```
{
    "batchcomplete": "",
    "continue": {
        "sroffset": 5,
        "continue": "-||"
    },
    "query": {
        "searchinfo": {
            "totalhits": 153887
        },
        "search": [
            {
                "ns": 0,
                "title": "Исследование",
                "pageid": 1503989,
                "size": 11327,
                "wordcount": 536,
                "snippet":     "окружающего
мира.          Такое          <span
class=\"searchmatch\">исследование</span>
может   иметь   практическое   применение.
Различают   эмпирическое   и   теоретическое
<span
class=\"searchmatch\">исследование</span>.
Оно строится следующим",
                "timestamp":      "2025-07-
16T10:39:35Z"
            },
            {
                "ns": 0,
                "title":    "Ультразвуковое
```

```
исследование",
                "pageid": 74119,
                "size": 73170,
                "wordcount": 4344,
                "snippet":  "Ультразвуково́е
<span
class=\"searchmatch\">иссле́дование</span>
(УЗИ),  сонографи́я  —  неинвазивное  <span
class=\"searchmatch\">исследование</span>
организма человека или животного с помощью
ультразвуковых волн. Физическая",
                "timestamp":      "2025-07-
16T10:14:36Z"
            },
            {
                "ns": 0,
                "title":       "Клиническое
исследование",
                "pageid": 50457,
                "size": 147036,
                "wordcount": 8940,
                "snippet":     "Клини́ческое
<span
class=\"searchmatch\">иссле́дование</span> —
научное                              <span
class=\"searchmatch\">исследование</span> с
участием людей, которое проводится с целью
оценки эффективности и безопасности нового
лекарственного",
                "timestamp":      "2025-07-
24T11:43:44Z"
            },
            {
                "ns": 0,
                "title":           "Научное
```

```
исследование",
                "pageid": 1312974,
                "size": 15564,
                "wordcount": 889,
                "snippet":  "знаний.  Виды
исследований:     Фундаментальное     <span
class=\"searchmatch\">исследование</span> —
теоретическое или экспериментальное научное
<span
class=\"searchmatch\">исследование</span>
основополагающих      явлений,      базовых
принципов",
                "timestamp":      "2025-05-
20T21:08:01Z"
            },
            {
                "ns": 0,
                "title": "Солнце",
                "pageid": 633,
                "size": 219617,
                "wordcount": 13086,
                "snippet":     "спектрометр
ультрафиолетового   диапазона.   Основной
задачей     Hinode     является     <span
class=\"searchmatch\">исследование</span>
активных  процессов  в  солнечной  короне  и
установление их связи со структурой",
                "timestamp":      "2025-07-
18T08:08:39Z"
            }
        ]
    }
}
```