Generating Synthetic Training Data for Named Entity Recognition With Large-Scale Models Integrating Wikidata and GPT

Adel BELBEKRI[1,†], Wissem BOUARROUDJ[1,*,†], Fouzia BENCHIKHA[1] and Zizette BOUFAIDA[1]

[1]*Lire Laboratory, Abdelhamid Mehri Constantine 2 University*

Abstract

Named Entity Recognition (NER) remains a critical task in Natural Language Processing, essential for identifying and classifying named entities within text data. Despite recent advancements, there is an ongoing need for diverse, high-quality datasets tailored to various languages, domains, and specific applications. This paper presents a novel approach to create a dataset for NER by leveraging Wikidata Knowledge Graph. We utilize the rich structured knowledge to extract entities and their associated types. These types undergo multi-categorization, providing a comprehensive representation of entity classifications. Additionally, we employ the GPT API for content generation, enhancing the dataset's richness and diversity. By integrating AI-driven content creation with structured knowledge from Wikidata, our approach offers an opportunity to refine NER models through access to structured knowledge and synthetic examples. The results highlight the diverse distribution of named entities across categories, emphasizing the importance of fine-grained categories for training robust models adaptable to various domains. Through this work, we aim to address gaps in NER dataset availability and contribute to developing more robust and accurate NER systems.

Keywords

Named Entity Recognition, Natural Language Processing, Synthetic text, AI-generated text, Knowledge Graph

1. Introduction

Named Entity Recognition (NER) is a fundamental task in Natural Language Processing (NLP) that involves identifying and classifying named entities within text data. Named entities refer to specific types of entities such as persons, organizations, locations, dates, events, and more. The primary objective of NER is to extract and categorize these named entities to facilitate various downstream NLP tasks and applications.

The origins of NER can be traced back to the early days of information extraction and text mining, where researchers sought to automate the process of identifying and extracting relevant information from unstructured text sources. Over time, NER has evolved into a critical component of many NLP systems and applications [1], playing a crucial role in tasks such as information retrieval [2], question answering [3], document summarization [4], and sentiment analysis [5].

Despite its advancements, NER still faces significant challenges due to the ambiguity and variability of named entities in natural language text [6]. For instance, the same entity can be referred to using different surface forms or aliases, and context plays a crucial role in disambiguating entities with multiple meanings. Additionally, named entities may exhibit complex structural and semantic relationships within the text, further complicating the accuracy of identification and classification.

Recent years have witnessed remarkable progress in NER due to the proliferation of large-scale annotated datasets, the development of sophisticated machine learning algorithms, and the availability of powerful computational resources. State-of-the-art NER models often leverage deep learning

RIF'24: The 13th Conference on Research in computIng at Feminine, May 20-21, 2024, Constantine, Algeria

*Corresponding author.

†These authors contributed equally.

adel.belbekri@univ-constantine2.dz (A. BELBEKRI); wissem.bouarroudj@univ-constantine2.dz (W. BOUARROUDJ); fouzia.benchikha@univ-constantine2.dz (F. BENCHIKHA); zizette.boufaida@univ-constantine2.dz (Z. BOUFAIDA)

0009-0008-3462-4256 (A. BELBEKRI); 0000-0002-0730-9495 (W. BOUARROUDJ); 0000-0002-4128-3620 (Z. BOUFAIDA)

architectures such as recurrent neural networks (RNNs) and transformers [7] to achieve impressive performance across various text domains and languages.

As artificial intelligence rapidly advances, language models have attained remarkable skill in generating persuasive, coherent texts. However, this burgeoning realm of AI-generated synthetic texts presents novel challenges for NLP, with NER bearing a significant brunt. This study addresses these challenges by exploring methodologies to improve NER systems' performance on synthetically generated texts.

We aim to determine optimal strategies for integrating structured knowledge from sources like Wikidata with synthetic data generation techniques using models such as GPT-3. Our approach seeks to enhance the resilience of NER models by providing them with diverse and contextually rich datasets that reflect real-world complexities.

The remainder of this paper is structured as follows: Section 2 provides a comprehensive background on NER; Section 3 surveys related work in the field; Section 4 details the construction of the dataset used in our study; and Section 5 concludes by summarizing our key findings and discussing avenues for future research.

2. Background

This section introduces key topics essential to understanding this paper. It covers Generative AI, Wikidata Knowledge graph and the SPARQL query language. These concepts serve as foundational knowledge for comprehending the context and methodology of the study.

2.1. Generative AI

Generative AI [8] refers to a class of artificial intelligence (AI) techniques and models designed to generate new data samples that resemble, or are indistinguishable from, examples in a given dataset. Unlike traditional AI systems focusing on classification or prediction tasks, Generative AI aims to create new content, such as images, text, audio, or video, that exhibits certain desired characteristics tailored to its intended purpose. These characteristics may include attributes like realism, creativity, coherence, relevance, sentiment, etc. depending on the specific goals and context of the generative AI system.

Generative AI models leverage techniques like deep learning, probabilistic modeling, and neural networks to learn the underlying patterns and structures of the training data and generate novel samples based on this learned knowledge. These models can be trained on large example datasets to capture complex relationships and generate realistic outputs.

Applications of generative AI span a wide range of domains, including creative content generation, data augmentation, image synthesis, text generation, and more. Generative AI has also found applications in art, design, entertainment, and virtual reality, where the ability to create new and diverse content is highly valued.

2.2. Wikidata Knowledge graph

Wikidata [9] is a collaborative knowledge base maintained by the Wikimedia Foundation. Launched in 2012, it is a centralized repository of structured data to support Wikimedia projects and external applications. Users contribute and edit data in a structured format, creating a comprehensive and multilingual knowledge base.

2.3. SPARQL query language

SPARQL [10] is a query language that retrieves and manipulates data stored in RDF (Resource Description Framework) format. RDF is a standard model for representing data on the web, often used to describe resources, their properties, and the relationships between them.

SPARQL provides a powerful and flexible way to query RDF data by expressing patterns of triples (subject-predicate-object statements) that match the desired information, making SPARQL suitable for various applications, including data integration, semantic web development, and linked data analysis.

2.4. Named entity recognition

NER is often described as a sub-task of information extraction in Natural Language Processing (NLP). It involves identifying and classifying named entities in text into predefined categories such as person names, organizations, locations, medical codes, time expressions, quantities, monetary values, and more. This definition emphasizes NER's role in transforming unstructured text into structured data, which is crucial for various applications like data analysis and knowledge graph construction.

NER serves as a bridge between unstructured text and structured data, enabling machines to sift through vast amounts of textual information and extract valuable data in categorized forms. This perspective highlights NER's utility in making data actionable for tasks like information retrieval and semantic search.

3. Related works

Synthetic data generation has emerged as a promising approach to address the challenges of creating high-quality datasets in various domains, including named entity recognition. This section reviews related works exploring different techniques for generating synthetic data applied to NER and similar tasks. While these studies highlight the growing interest in these methods and their potential advantages, there remain some limitations and areas for improvement that our work aims to address.

Libbi et al. [11] generate synthetic Electronic Health Records using language models. The authors use large language models (LSTMs and GPT-2) trained on real EHR data to generate synthetic EHR text. By explicitly adding in-text annotations to the training data, the language models learn to produce artificial text automatically annotated for downstream NER tasks. The experiments show that augmenting real data with synthetic data can improve the recall and coherence of the data. However, their approach relies on having high-quality annotated seed data, which may not always be available, especially for low-resource domains or languages.

Samudra et al. [12] generate synthetic data to develop and test entity recognition algorithms appropriate for big data. They proposed a simulation model that can generate name-like vectors. This approach takes a dataset of real name strings and computes the pairwise dissimilarities between them. Then, it uses MDS to map these name strings into a lower-dimensional Euclidean vector space, while attempting to preserve the pairwise dissimilarities as Euclidean distances between the vector representations (referred to as name-like vectors). Additionally, it analyzes whether these name-like vectors follow a multivariate normal distribution. If so, it estimates the mean vector and covariance matrix parameters. Afterwards, it generates new synthetic name-like vectors by sampling from this estimated multivariate normal distribution, efficiently producing large volumes of name-like vector data.

Kuo [13] introduces a comprehensive workflow for synthesizing authentic insurance datasets utilizing a neural network-based generative model known as CTGAN. The authors initiate by training the CTGAN architecture on a proprietary insurance dataset, allowing it to effectively capture the underlying data distribution. Subsequently, synthetic tabular data samples are generated from the trained CTGAN model. Following this, dataset-specific pre-processing and post-processing transformations are applied to uphold the consistency and domain relevance of the synthetic data. The authors evaluate the proposed workflow on two publicly available insurance datasets for general insurance pricing and life insurance shock lapse modeling. They assess the quality of the synthesized data by comparing the efficacy of predictive models trained on real vs synthetic data, analyzing variable distributions, and examining the stability of model parameters fitted on the synthetic data. While valuable for tabular data synthesis, this approach may struggle to generalize to unstructured text common in NER tasks.

Additionally, certain approaches leverage knowledge graphs to construct datasets. For instance, Specht Menezes et al. [14] propose a method to automatically generate a massive labeled dataset

for NER by exploiting structured data from DBpedia and Wikipedia knowledge graphs. First, they extract data from DBpedia to obtain a list of entities (people, organizations, locations) along with their names/aliases and categories. Then, they extract text data from Wikipedia articles. Therefore, they link the DBpedia entities to mentions in the Wikipedia text by exact string matching of the entity names/aliases. Finally, they preprocess and tokenize the text to annotate the identified entity mentions. This approach generates a dataset called SESAME, which serves to enhance the development of more robust NER predictors.

While the preceding studies have undoubtedly made significant contributions, the field still faces challenges in generating diverse and precisely annotated synthetic NER datasets. This is particularly evident in addressing the complexities posed by large language models and their synthetic outputs. These challenges stem from the limitations of existing methods in generating comprehensive datasets that capture the complexity and nuances of real-world data. To address these challenges, we propose a novel approach that combines approaches for creating synthetic data with powerful language models like GPT-3 to generate texts, with approaches that explore knowledge graphs such as Wikidata to retrieve relevant data. This hybrid approach aims to leverage the strengths of both techniques, resulting in more diverse and contextually rich datasets for NER tasks. Furthermore, by ensuring the multi-categorization of entities and balanced category representation, our dataset aims to mitigate potential biases and produce well-rounded NER training data. We also explore different data formats to enable use cases beyond just NER.

4. Dataset construction

We adopted an innovative approach that combines a random sampling of named entities from the Wikidata knowledge graph and example generation to create a high-quality dataset for the named entity recognition task. The data are then formatted using CONLL2003 and JSON formats. Figure 1 depicts constructing this dataset. The details of each phase are explained in the following subsections.

4.1. Random entity retrieval

In our process, we used the random functionality provided by Wikidata to retrieve named entities randomly. This approach ensured that our entity selection was unbiased and representative of the diverse range of entities present in the knowledge base. However, we recognized that simply retrieving entities randomly might not capture the full breadth of information available. Many entities in Wikidata are associated with multiple categories, reflecting their multifaceted nature and relationships to various domains. To address this, we implemented an additional step in our process.

For each randomly extracted entity that belonged to multiple categories, we meticulously collected and retained all relevant categories associated with that entity. This comprehensive approach ensured that our analysis and subsequent representations accurately reflected the complete context related to each multi-categorized entity.

However, we observed a significant variation in the number of randomly selected examples across different categories (Table 1). To mitigate the potential bias or overfitting that could arise from this imbalanced distribution, we propose employing a combination of undersampling and oversampling techniques during the sentence generation phase. For over-represented categories with a disproportionately high number of randomly selected examples (e.g., ORGANISATION with 111,510 examples), we can perform undersampling by randomly selecting a subset of the examples to be used for sentence generation. The number of examples to be retained is mentioned in Table 1. For under-represented categories with a relatively low number of randomly selected examples (e.g., PLACES with 12,862 examples), we can employ oversampling techniques to increase the number of examples used for sentence generation. One approach could be to perform data augmentation by generating multiple sentences for each example in the under-represented category, effectively increasing the representation of these categories in the final dataset. By applying undersampling for over-represented categories and oversampling for under-represented categories, we can achieve a more balanced distribution of

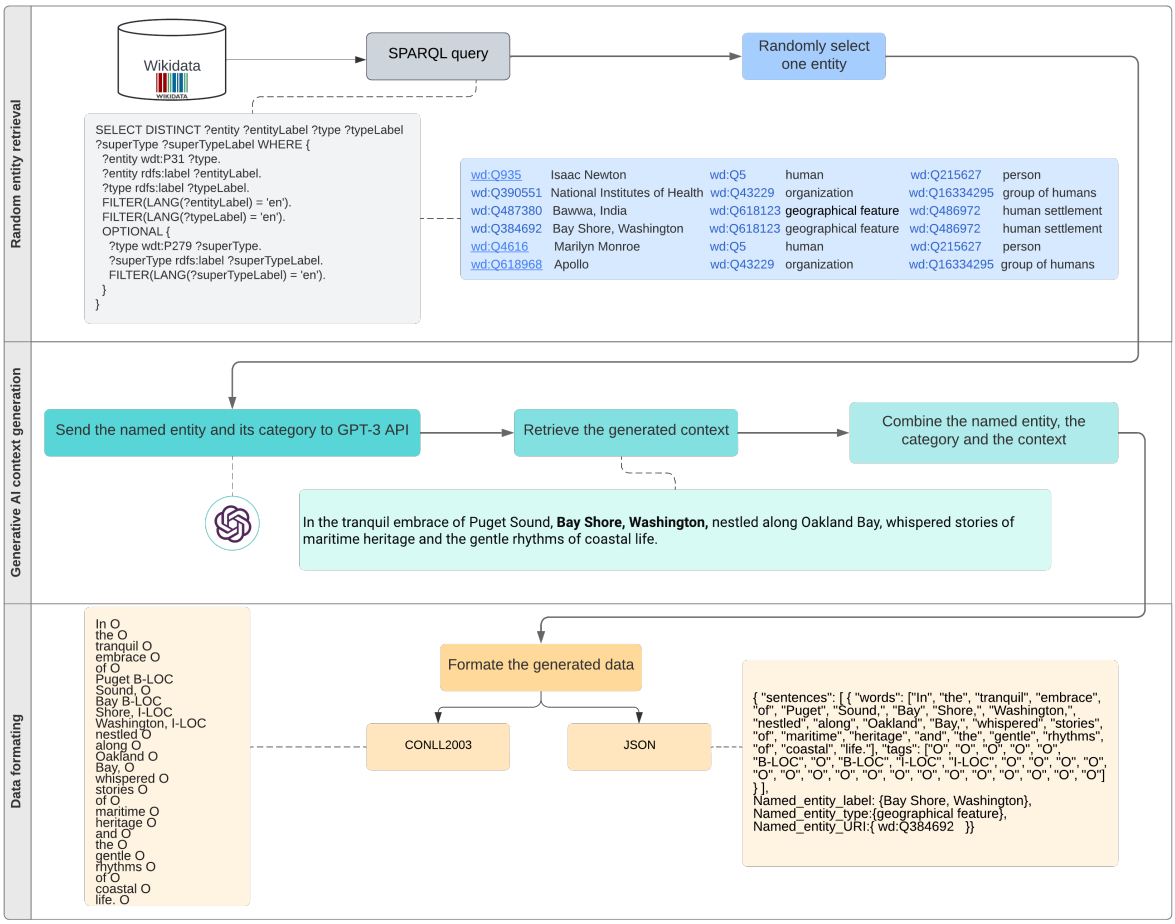

Figure 1: Dataset construction process

named entities across categories in the final dataset. This balanced representation is crucial for training robust and generalizable named entity recognition models that perform well across various domains and contexts, without being biased towards or overfitting on any particular category.

To facilitate the exploration and querying of the Wikidata knowledge graph, we employed a SPARQL query presented in Listing 1. This query enabled us to navigate the intricate web of relationships and retrieve the desired information from the knowledge graph effectively.

```
SELECT DISTINCT ?entity ?entityLabel ?type ?typeLabel
?superType ?superTypeLabel WHERE {
  ?entity wdt:P31 ?type.
  ?entity rdfs:label ?entityLabel.
  ?type rdfs:label ?typeLabel.
  FILTER(LANG(?entityLabel) = 'en').
  FILTER(LANG(?typeLabel) = 'en').
  OPTIONAL {
    ?type wdt:P279 ?superType.
    ?superType rdfs:label ?superTypeLabel.
    FILTER(LANG(?superTypeLabel) = 'en').
  } }
```

Listing 1: SPARQL query

In this query:

- ?entity represents the entity.

- ?entityLabel represents the English label of the entity.
- ?type represents the type of the entity.
- ?typeLabel represents the English label of the type.
- ?superType: Represents the superior class (if available) of the type.
- ?superTypeLabel: Represents the English label of the superior class.
- wdt:P31 is used to specify the "instance of" property. It is used to specify the type or class that an entity belongs to. Essentially, it indicates what category or class an item falls under in Wikidata's ontology.
- FILTER(LANG(?entityLabel) = 'en') and FILTER(LANG(?typeLabel) = 'en') Ensure that only English labels are retrieved for both entities and types.
- OPTIONAL ... : Defines an optional block where the superior class (?superType) of each type (?type) is retrieved using the wdt:P279 property (subclass of). If a superior class exists, its label (?superTypeLabel) is also retrieved.
- FILTER(LANG(?superTypeLabel) = 'en'): Ensures that only English labels are retrieved for the superior class.

For each retrieved entity, we considered two categories: one fine-grained, the category selected by the "instance of" property, and one coarse-grained selected using the superclass of the chosen category. By focusing on these significant distinctions, we aim to ensure a clearer and more precise categorization of the entities within our dataset. Fine-grained categories allow for detailed classification, capturing nuanced differences between entities, while coarse-grained categories provide broader groupings, offering a high-level overview of the dataset's composition. This approach enables us to balance granularity and comprehensiveness, facilitating effective organization and analysis of the data according to our research objectives.

4.2. Generative AI context generation

We used the powerful GPT-3 language model [15] with a Python code to automatically interface with the API. To generate diverse textual examples containing the named entities extracted directly from Wikidata. The GPT-3 API enabled us to create realistic synthetic text examples by providing the named entity and associated type(s) from Wikidata as input prompts. These prompts were carefully structured to include placeholders for the entity and its type(s), guiding GPT-3 to generate coherent text appropriately incorporating the given information.

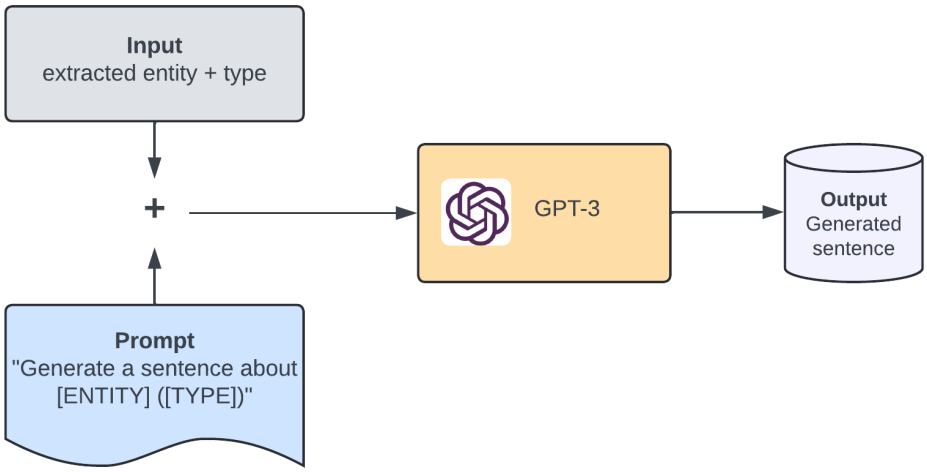

Figure 2: GPT-3 samples generation

The prompt "Generate a sentence about [ENTITY] ([TYPE])" allows GPT-3 to produce a relevant sentence mentioning the specific named entity while contextualizing it based on the provided type

(Figure 2). This approach leveraged GPT-3's language generation capabilities to create diverse syntactic examples spanning various contexts, all while ensuring the named entities were seamlessly integrated into the generated text. After generating synthetic text examples, we retained the Wikidata category used in the prompt for each named entity present. This category was used to tag the corresponding entity in the generated text, following the standard IOB (Inside, Outside, Beginning) format for named entity recognition. Tokens belonging to an entity were tagged with the prefix B- or I- followed by the category. Other tokens were tagged with 'O'. Additional details on this tagging format are provided in the following subsection. By harnessing the structured knowledge from Wikidata and the powerful text generation of GPT-3, we could construct a rich dataset of synthetic examples valuable for training named entity recognition large-scale models.

4.3. Format choice

We adopted the CONLL2003 and JSON formats for several reasons. Firstly, the CONLL2003 format is widely used in academic research and industry, facilitating comparison and reproducibility of results across different models. Additionally, this format provides a simple tabular representation of the data, with columns dedicated to words, named entity tags, and other relevant information, making it a convenient choice for data storage and handling.

On the other hand, the JSON format was chosen for its versatility and ease of use. As a widely supported structured data format across many programming languages and libraries, it offers flexibility in data representation, allowing the storage of additional information or metadata associated with the named entities. In our case, we leverage this flexibility to include relevant information about the label of the named entity and its corresponding URI in Wikidata. By incorporating this additional metadata, our dataset can be used for multi-purpose tasks beyond just named entity recognition, such as entity linking [16], where the ability to map named entities to their unique identifiers in a knowledge base is essential.

For both the CONLL2003 and JSON formats, the named entity tags are provided using the IOB (Inside, Outside, Beginning) format, efficiently representing named entity spans within text sequences. Each token (word) in the text is assigned a tag indicating its position relative to a named entity. The possible tags are:

- O (Outside): This token is not part of a named entity.
- B-[TYPE] (Beginning): This token marks the beginning of a named entity of the specified type (e.g., B-PER for a person entity).
- I-[TYPE] (Inside): This token is inside a named entity of the specified type, following the beginning token.

Listing 2: CONLL2003 example

```
In              O
the             O
tranquil        O
embrace         O
of              O
Puget           B-LOC
Sound,          O
Bay             B-LOC
Shore,          I-LOC
Washington,     I-LOC
nestled         O
along           O
Oakland         O
Bay,            O
whispered       O
stories         O
```

```
        of              O
        maritime        O
        heritage        O
        and             O
        the             O
        gentle          O
        rhythms         O
        of              O
        coastal         O
        life.           O
```

Listing 3: JSON example

```
{ "sentences":
    [
    { "words":
        ["In", "the", "tranquil", "embrace", "of", "Puget", "Sound,", "Bay", "Shore,", "
            Washington,", "nestled", "along", "Oakland", "Bay,", "whispered", "stories",
             "of", "maritime", "heritage", "and", "the", "gentle", "rhythms", "of", "
            coastal", "life."],
    "tags":
        ["O", "O", "O", "O", "O", "B-LOC", "O", "B-LOC", "I-LOC", "I-LOC", "O", "O", "O
            ", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O"]
    }
    ],
"Named_entity_label":{Bay Shore, Washington},
"Named_entity_type":{geographical feature},
"Named_entity_URI":{ wd:Q384692 }
}
```

By adhering to the IOB format, our dataset provides a standardized and well-established way of representing named entity annotations. , ensuring seamless integration and compatibility with existing NER systems and pipelines. Furthermore, the IOB format allows for efficient processing and evaluation of named entity recognition models, as it enables straightforward computation of metrics such as precision, recall, and F1-score at the entity level. The following Listings depicts samples of the resulting dataset in the CONLL2003 format (Listing 2) and JSON format (Listing 3).

4.4. Key statistics of the generated dataset

Table 1 provides an overview of the distribution of named entities across different categories in the generated dataset. One notable observation is the diversity of sub-categories (fine-grained) covered in the dataset. These categories are selected based on the coarse-grained categories from a previous work [17]. This diversity is essential for training robust and generalizable named entity recognition models that perform well across various domains and contexts.

Another interesting aspect is the varying number of subcategories within each main category that reflects the inherent complexity and granularity of different domains and the level of detail captured in the Wikidata knowledge graph.

The distribution of randomly selected examples across categories varies significantly. This distribution may reflect the underlying distribution of entities in the knowledge graph. This variation is adjusted in the sentence generation phase to avoid any potential unbalancing in the representation of the named entities that could produce biases or overfitting in the model training.

5. Conclusion

This paper presented a novel approach to constructing a high-quality dataset for training and evaluating named entity recognition models on large-scale data. Our methodology leverages the strengths of

Table 1
Key statistics of the generated dataset

Categorie	# sub-categories	# potential named entites	# of randomly selected examples	# of generated sentenses
PERSON	345	11281499	40710	122130
ORGANISATION	945	4718205	111510	111510
PLACES	109	13302073	12862	128620
CREATIVE	110	14569572	12980	129800
PRODUCT	449	13148658	52982	158946
EVENT	283	1273189	33394	133576
NATURAL	92	7341422	10856	108560
LANGUAGE	94	21661	11092	110920
SPORT	178	30743	21004	126024
FOOD	2935	20471	20471	122826

knowledge graphs, represented by Wikidata, and state-of-the-art language models like GPT-3.

By randomly extracting named entities from Wikidata and collecting their associated categories, we ensured an unbiased and comprehensive representation of entities across various domains. To maintain balanced category representation, we performed targeted extractions when necessary, mitigating potential biases and enabling the development of robust and generalizable NER models.

Our innovative approach addresses the need for diverse, high-quality datasets tailored to the rapidly evolving landscape of NER tasks, particularly in the context of large-scale data and AI-generated content. By combining knowledge graphs, language models, and careful data curation, we have created a valuable resource that can drive progress in developing more robust and accurate named entity recognition systems.

This paper lays the foundation for an in-depth exploration and expansion into a comprehensive journal article, in the future work we aim to evaluate the performance of NER models trained on our dataset across a range of real-world applications and domains, further validating the effectiveness of our methodology.

Declaration on Generative AI

During the preparation of this work, the authors used Perplexity in order to: Grammar and spelling check. After using these tool, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] Z. Nasar, S. W. Jaffry, M. K. Malik, Named entity recognition and relation extraction: State-of-the-art, ACM Computing Surveys (CSUR) 54 (2021) 1–39.

[2] N. Perera, M. Dehmer, F. Emmert-Streib, Named entity recognition and relation detection for biomedical information extraction, Frontiers in cell and developmental biology 8 (2020) 673.

[3] F. Zhu, W. Lei, C. Wang, J. Zheng, S. Poria, T.-S. Chua, Retrieving and reading: A comprehensive survey on open-domain question answering, arXiv preprint arXiv:2101.00774 (2021).

[4] G. Riccio, A. Romano, A. Korsun, M. Cirillo, M. Postiglione, V. La Gatta, A. Ferraro, A. Galli, V. Moscato, Healthcare data summarization via medical entity recognition and generative ai (2023).

[5] Y. Ma, Y. Zhang, A. K. Sangaiah, M. Yan, G. Li, T. Wang, Active learning for name entity recognition with external knowledge, ACM Transactions on Asian and Low-Resource Language Information Processing (2023).

[6] W. Bouarroudj, Z. Boufaida, L. Bellatreche, Named entity disambiguation in short texts over knowledge graphs, Knowledge and Information Systems 64 (2022) 325–351.
[7] B. Jehangir, S. Radhakrishnan, R. Agarwal, A survey on named entity recognition—datasets, tools, and methodologies, Natural Language Processing Journal 3 (2023) 100017.
[8] P. Eigenschink, T. Reutterer, S. Vamosi, R. Vamosi, C. Sun, K. Kalcher, Deep generative models for synthetic sequential data: A survey, IEEE Access (2023).
[9] A. Waagmeester, G. Stupp, S. Burgstaller-Muehlbacher, B. M. Good, M. Griffith, O. L. Griffith, K. Hanspers, H. Hermjakob, T. S. Hudson, K. Hybiske, et al., Wikidata as a knowledge graph for the life sciences, Elife 9 (2020) e52614.
[10] J. Pérez, M. Arenas, C. Gutierrez, Semantics and complexity of sparql, ACM Transactions on Database Systems (TODS) 34 (2009) 1–45.
[11] C. A. Libbi, J. Trienes, D. Trieschnigg, C. Seifert, Generating synthetic training data for supervised de-identification of electronic health records, Future Internet 13 (2021) 136.
[12] S. Herath, M. Roughan, G. Glonek, Generating name-like vectors for testing large-scale entity resolution, IEEE Access 9 (2021) 145288–145300.
[13] K. Kuo, Generative synthesis of insurance datasets, arXiv preprint arXiv:1912.02423 (2019).
[14] D. Menezes, R. Milidiu, P. Savarese, Building a massive corpus for named entity recognition using free open data sources, in: 2019 8th Brazilian Conference on Intelligent Systems (BRACIS), IEEE, 2019, pp. 6–11.
[15] L. Floridi, M. Chiriatti, Gpt-3: Its nature, scope, limits, and consequences, Minds and Machines 30 (2020) 681–694.
[16] W. Bouarroudj, Z. Boufaida, L. Bellatreche, Welink: a named entity disambiguation approach for a qas over knowledge bases, in: Flexible Query Answering Systems: 13th International Conference, FQAS 2019, Amantea, Italy, July 2–5, 2019, Proceedings 13, Springer, 2019, pp. 85–97.
[17] A. Belbekri, F. Benchikha, Y. Slimani, N. Marir, Socialner2. 0: A comprehensive dataset for enhancing named entity recognition in short human-produced text, Intelligent Data Analysis (2024) 1–25.