Sensitivity Analysis of Named Entity Extraction based on Deep Learning

Lea Roj
lea.roj1@student.um.si
Faculty of Electrical
Engineering and Computer Science,
University of Maribor
Koroška cesta 46
SI-2000 Maribor, Slovenia

Štefan Kohek
stefan.kohek@um.si
Faculty of Electrical
Engineering and Computer Science,
University of Maribor
Koroška cesta 46
SI-2000 Maribor, Slovenia

Aleksander Pur
aleksander.pur@policija.si
Ministry of the Interior,
Štefanova ulica 2
SI-1000 Ljubljana, Slovenia

Niko Lukač
niko.lukac@um.si
Faculty of Electrical
Engineering and Computer Science,
University of Maribor
Koroška cesta 46
SI-2000 Maribor, Slovenia

ABSTRACT

In the ever-evolving landscape of data visualization and extraction, deep learning techniques have become increasingly pivotal. Addressing this, our paper conducts a sensitivity analysis of Named Entity Extraction on text related to the infamous Panama Papers. We combine the Rebel and Natural Language Processing models to extract entities and relations. To visualize the extracted knowledge, we employ the NetworkX library to transform this data into intuitive graphs. These are then compared to a predefined expected graph to assess the influence of various parameters during the creation process. To ensure an accurate comparison, the graphs are transformed into vectors using the `Graph2Vec` method after undergoing pre-processing tasks, such as the removal of self-loops, isolated nodes, and node renaming. The similarity with our benchmark graph is determined using Euclidean distance metrics. Results highlight the influence of span length, length penalty, and the number of beams on graph generation. Notably, span length emerges as the most impactful factor, in determining graph detail and complexity.

KEYWORDS

Entity Extraction, Relation Extraction, Graph Generation, Embeddings, Graph Similarity

1 INTRODUCTION

Graphs provide an intuitive, more appealing visualization of data, revealing patterns and correlations that are not instantly visible from raw data. An alternative to graphs are relational and non-relational databases. Relational databases store data in structured tables and use primary and foreign keys to interconnect data. Through these connections, relational databases can generate valuable insights by merging tables [10]. On the other hand, non-relational databases, often called NoSQL databases [18], are non-tabular databases and excel in horizontal scalability, allowing more machines to be easily integrated. Their maintenance is cheap as they do not depend on expensive hardware and the accommodation of changes is easier. Moreover they are designed to store simple data structures as a key and a value.

Graphs, relational databases, and NoSQL databases are fundamental in data management but differ significantly. In graphs, data is stored as nodes, edges, and properties, offering flexibility when adding new relationships or nodes. In contrast, relational databases use tables with rows and columns, potentially requiring substantial redesigns when introducing new data relationships and data models. On the other hand NoSQL databases are more adaptable and can easily handle changes and a variety of data types, making them a scalable solution. While graphs employ query languages like Neo4j's Cypher [20], relational databases predominantly use SQL [14], and NoSQL databases use a variety of query languages. Extracting semantic relationships using relational databases for documents is challenging. Representing data with graphs emerges as an effective solution. A primary challenge is determining the optimal parameters for graph construction, which is the focus of our paper.

The concept of a "Named Entity" and relation extraction was first put forward during the Sixth Message Understanding Conference [5] in 1995. Recent advancements were described by Lung-Hao et al. in [11]. Insights from the publications indicate the development and evaluation of the capability of a Chinese healthcare NER recognizer. The best results were achieved by the MIGBaseline team using the BERT-BiLSTM-CRF model [3].

While Named Entity Extraction (NEE) identifies entities, Relation Extraction (RE) seeks to determine relationships between these identified entities. In the article from 2023, Moscato et al. [15] presented a multi-task framework for biomedical relation extraction using a transformer-based model trained on three publicly available datasets related to drug, protein, and medical relationships. Expanding on this theme of advancements in RE, the Collaborative Oriented Relation Extraction (CORE) system offers another noteworthy contribution. Introduced in 2023 by Marchesin et al. [13],

DOI: https://doi.org/10.26493/scores23.12

the CORE system was specifically designed to extract multi-aspect relationships from scientific texts.

In this paper, we introduce a novel workflow for testing hyperparameters and performing sensitivity analysis in the context of entity resolution, using Wikipedia summaries. Our approach employs Term Frequency - Inverse Document Frequency (TF-IDF) vectorization combined with cosine similarity in the specific context of entity resolution. While contemporary methods, such as Word2Vec [4] and BERT, provide more advanced ways to compute word similarity by capturing semantic meanings of the words, we opted to use the TfidfVectorizer [16] to transform textual data into a TF-IDF representation. Even though it is not the state-of-the-art it is effective and simple to use, due to its simplicity.

We perform a detailed comparison by converting these summaries into TF-IDF vector representations and subsequently calculating their cosine similarity. This strategy enables us to efficiently describe entities that various names could refer to while essentially denoting the same concept. Furthermore, by embedding the cosine similarity as an attribute of the relationship, we introduce a direct metric for quantifying the relevance between linked entities. Our method combines advanced NLP methods. The process begins with tokenization to prepare the text, followed by model-based generation to structure the information. A distinctive feature of our approach is the overlap strategy we have adopted in tandem with relation extraction. By segmenting long texts with overlaps, we maintain context and ensure relationships are not missed or fragmented.

The structure of this paper spans four sections. After the introduction, the following section delineates the methods and functions employed. The third section presents the results, including a sensitivity analysis concerning different parameters and their impact on NEE. The last section concludes the paper.

2 METHODOLOGY

The following subsections will provide a comprehensive breakdown of important methods implemented in the project.

2.1 Named entity extraction

Firstly, the input text is initially processed using the `en_core_web_lg`, an English language model, from the `spacy` library [7]. This pretrained model supports various NLP tasks such as NER, RE, and tokenization. Tokenization involves separating the raw text into smaller units known as tokens. Each token ID represents a unique word in the text, while same words share the same ID. By dividing text into manageable spans, determining the start and end tokens of those spans based on span and overlap length, this process converts the human-readable text into a format that machine-learning models can understand and prepares the text for further handling.

The span length determines the size of the processed text segments and is associated with the number of words. Opting for a lower number results in a more detailed graph, because more individual spans are analyzed. However, selecting an optimal span length is crucial because if the value is too small, the text can become too fragmented, rendering the represented nodes meaningless. The influence of span length is exemplified in Fig. 1. The graph in Fig. 1a contains 9 entities, whereas the graph in Fig. 1b has 35 entities. As can be seen, in graph Fig. 1a, an increased span length results in fewer spans being analyzed. Consequently, it may overlook some entities or relations that might be present in smaller spans.

On the other hand, the overlap length ensures a certain degree of continuity between consecutive text segments, preserving context. Through this overlap, we minimize the risk of losing essential contextual information at segment boundaries. The value represents the number of words from the previous span, that are also used at the beginning of the next one.

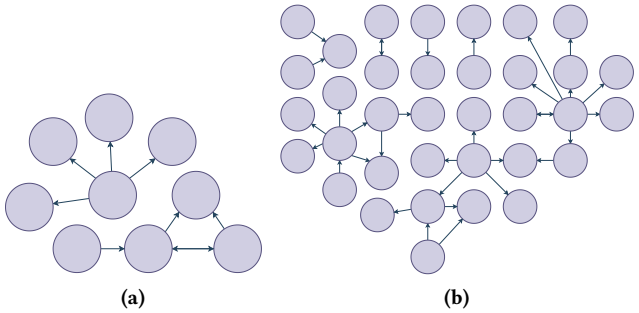

Figure 1: Impact of span length on graph generation when a) Span length = 120 and b) Span length = 50.

For Named Entity Recognition (NER), RE, and knowledge base creation [19] the rebel-large [9] model is used. It is a sequence-to-sequence (seq2seq) architecture, which is based on BART [12]. The seq2seq design enables the model to perform end-to-end RE tasks. It takes an input sequence, processes it, and produces an output sequence. In the context of "rebel-large," these output sequences represent triplets, where each triplet consists of a "head," a "tail," and the "relation" between them. The rebel-large model is prevalent in deep learning applications as it can identify and categorize named entities from raw text [1]. In conjunction with the model, initializing a corresponding tokenizer is critical, as it transforms raw text into a format the model can effectively process.

Handling vast amounts of data effectively is critical in optimizing the performance of deep learning models. The approach we took enables models to manage varying sizes of data by configuring specific parameters, such as input text, span length, overlap length, maximum length, length penalty, number of beams, number of return sequences, and the tokenizer's maximum length. Length penalty plays a pivotal role in sequence range determination. A positive value encourages longer sequences, a zero value maintains neutrality, and a negative value leans towards shorter sequences [21]. The number of beams influences the quality of the results by determining how many top sequences undergo exploration at each decision point or iteration during the search process. By employing the beam search algorithm, the number of return sequences determines how many top sequences are retrieved. The parameter maximum length defines the upper limit for the length of processed sequences, while the maximum length tokenizer indicates the maximum allowed length of text following tokenization. Given

the length of the processed text, span, and overlap, the text is divided into manageable segments, with each segment subsequently undergoing tokenization.

2.2 Relation Classification (RC)

When provided with a span of text, which was pre-processed using a NLP function, the rebel-large model predicts potential relationships, which are represented in token sequences. These sequences are subsequently decoded back into textual representations using the tokenizer. Within these textual representations, specific tokens ("<triplet>", "<subj>", "<obj>") help delineate and structure the relationship. We represent these potential relationships with "head", "tail", and "type". The "head" corresponds to the starting entity, while the "tail" refers to the ending entity of the relationship. The "type" characterizes how the two entities are related. For instance, as illustrated in Fig. 2, the "type" serves as the label on the relationship's edge.

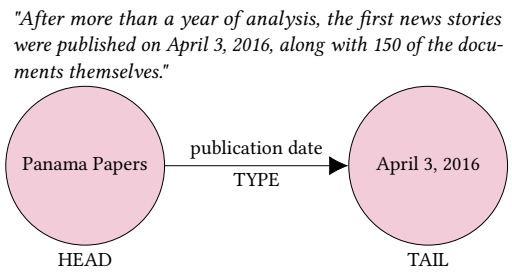

Figure 2: Illustration of relationship extraction from text.

2.3 Entity Filtering and Normalization

Entities undergo validation on Wikipedia via an HTTP GET request targeting a specific entity through Wikipedia's API. In response, the API returns data in JSON format, which includes the official title or name of the entity as recognized by Wikipedia, a link leading to the detailed Wikipedia page for the entity, and a brief overview of the entity. If entity data is found, then the entity title is the same as the one on the Wikipedia page. On the other hand, if data is not found, then it uses the provided entity name.

Cosine similarity is a common metric for measuring document similarity. By comparing summaries of entities from Wikipedia, we determine the degree of similarity between them. Using the TfidfVectorizer [16], textual data is transformed into a TF-IDF representation. Subsequently, the function `find_similar_entity` computes the cosine of the angle between the given summary and the Wikipedia summary of each stored entity, and returns the entity with the highest similarity score. Both summaries are vectors, representing the weighted word frequencies in a document [8]. A cosine similarity of 1 indicates that the vectors are identical, while a cosine similarity of 0 indicates that the vectors are completely dissimilar. Using Eq. 1 we solved the challenge of multiple entities sharing common names or terms. For example "Napoleon Bonaparte" and "Napoleon I" are two names for the same entity. Cosine Similarity is calculated as:

$$\frac{\vec{A} \cdot \vec{B}}{||\vec{A}||||\vec{B}||}, \tag{1}$$

where $\vec{A}$ and $\vec{B}$ present TF-IDF vectors of each summary.

2.4 Similarity calculation

The objective was to identify graphs resembling a predefined or expected graph. Before calculating similarity using embeddings from Graph2Vec [17], we preprocess each graph to ensure uniformity. This process includes removing self-loops, eliminating isolated nodes, and renaming nodes based on their sorted order in the node list of the graph. Subsequently, the `get_embedding` function obtains the desired embeddings.

Representing each graph as a vector in high-dimensional space enables more efficient graph comparison and analysis. Embeddings can be plotted in a coordinate system, where the proximity between points corresponds to their similarity. The Euclidean distance offers a measure of this distance by calculating the vector length between two points [2].

Using the graph embeddings, we define the Euclidean distance as:

$$d_i = \sqrt{\sum_{j \in I-i} (A_j - B_{i,j})^2}, \tag{2}$$

where A_j denotes the expected graph's embedding and $B_{i,j}$ represents the embeddings from other graphs. I is the set of all dimensions or indices in the high-dimensional space.

3 RESULTS

The analysis was conducted using a range of fixed and variable parameters. Those were chosen by heuristically testing parameters, based on which had the greatest influence on the graph generation and its quality. The variable parameters include span length, length penalty, and number of beams. On the other hand, the fixed parameters were set as:

- overlap length = 25
- maximum length = 130
- tokenizer's maximum length = 512

Table 1: Parameter configurations and similarity.

Span length	Length penalty	Number of beams	Similarity
90	-3	12	81.05%
60	-5	6	68.46%
50	0	6	66.73%
60	0	8	54.63%
90	0	6	52.20%
40	-5	6	46.60%
60	0	12	46.57%
50	5	12	45.88%
60	-1	6	41.78%
60	1	6	36.94%
40	-3	6	36.28%
40	1	6	35.12%
60	3	6	29.66%
60	5	6	21.40%
30	0	6	0.00%

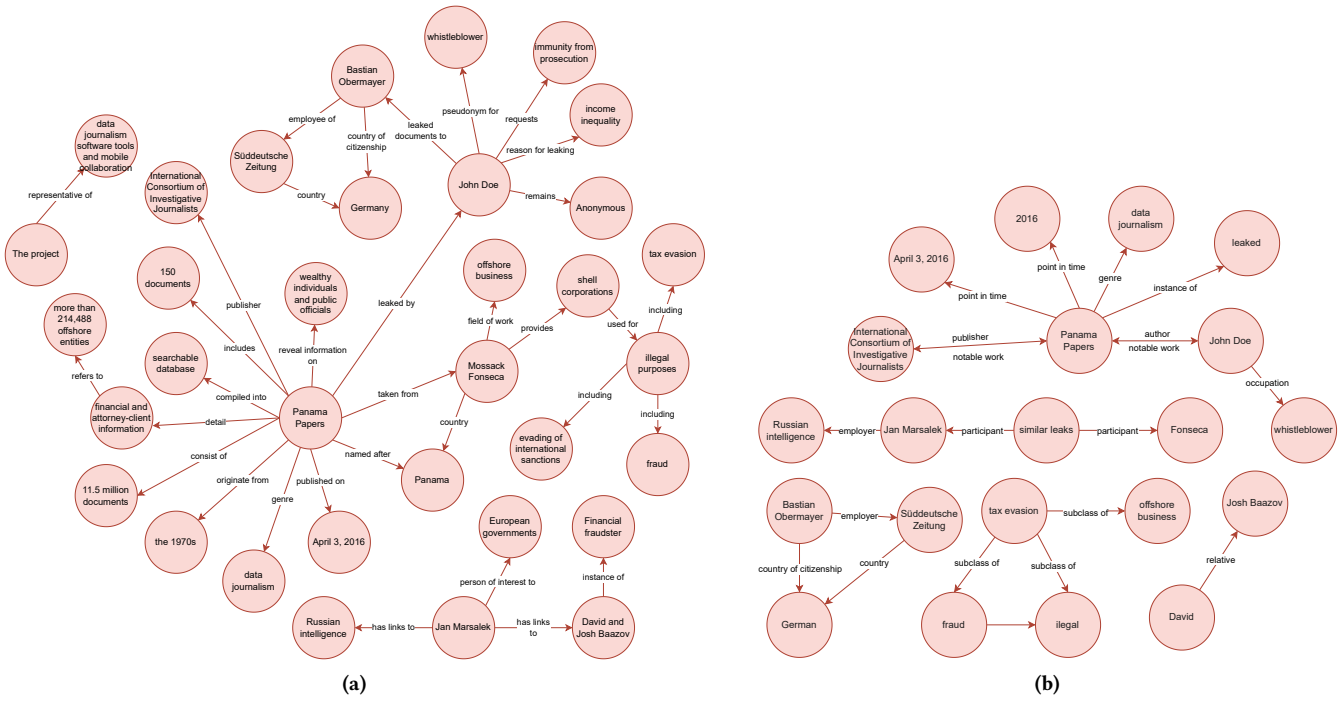

Figure 3: Graph comparison: a) Ground truth and b) Graph with similarity score 81.05%.

By exploring given variations of different parameters, we aim at analyzing sensitivity impact of them on generated graphs. In the table 1 are represented main results in terms of similarity. The table doesn't show systematic way of changing the parameters values because there was a lot of duplication and minor differences. Based on the Panama Papers text, the predefined graph in Fig. 3a consists of 34 nodes and 33 edges. This graph was constructed manually and contains a detailed representation of the data. All graphs were visualized using NetworkX [6].

The analysis was conducted on a system with Intel Core i7-13700K CPU, GeForce RTX 3070 Ti GPU, and 32 GB of RAM.

Upon comparison of the most similar graph (Fig. 3b) and the predefined graph (Fig. 3a), distinct parallels emerge. Both contain pivotal entities such as "Panama Papers", "John Doe", "Bastian Obermayer", and "Jan Marsalek". Other entities are mostly grouped around those main nodes. However, distinctions are evident. The generated graph explores diverse relationship types, featuring categories like "relative" and "subclass of". In contrast, the ground truth graph has more descriptive relationships, including terms such as "originate from", "compiled into", "leaked documents to", "reason for leaking", and "representative of". The predefined graph also contains more detailed entities, for example "11.5 million documents" and provides specific information behind John Doe's leaks ("income inequality"). The generated graph has a broader focus, considering relationships between various entities and their relevance to the Panama Papers. Meanwhile, the predefined graph delves deeper into specific details, like the services "Mossack Fonseca" provides and the nature of "shell corporations".

4 CONCLUSION

Named entity extraction is essential for extracting specific details from large sets of documents, making them clearer and easier to understand. Such extraction techniques can be used in journalism or intelligence systems to automatically identify pivotal events and their associated key entities. The proposed solution is presented on one dataset, but the same methodological approach could be applied to different data. This forms the foundation for the use of optimization algorithms and for selecting optimal hyper-parameters of the graph. The generated graphs, as observed, offer a panoramic view of the textual data and a quick overview, in contrast to predefined graph that delves into specifics.

Future advancements in this domain hold great promise. Considering a potential node coloring based on NER categories and introduction of more detailed relations, which would provide us with more information. For instance, a model trained on medical data, would not perform well on financial data without additional training. Therefore, training models to specific domains can vastly improve extraction accuracy. Ultimately, while current NER systems provide invaluable insights and streamline information processing, continuous improvement and domain-specific adaptations will be the cornerstone for achieving remarkable precision and clarity in the future.

ACKNOWLEDGMENTS

The authors acknowledge joint financial support from the Slovenian Research Agency and Slovenian Ministry of the Interior (target research program No. V2-2260).

REFERENCES

[1] Tareq Al-Moslmi, Marc Gallofré Ocaña, Andreas L. Opdahl, and Csaba Veres. 2020. Named Entity Extraction for Knowledge Graphs: A Literature Overview. *IEEE Access* 8 (2020), 32862–32881. https://doi.org/10.1109/ACCESS.2020.2973928

[2] Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*. Vol. 4. Springer.

[3] Zhenjin Dai, Xutao Wang, Pin Ni, Yuming Li, Gangmin Li, and Xuming Bai. 2019. Named Entity Recognition Using BERT BiLSTM CRF for Chinese Electronic Health Records. In *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. 1–5. https://doi.org/10.1109/CISP-BMEI48845.2019.8965823

[4] Yoav Goldberg and Omer Levy. 2014. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722* (2014).

[5] Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference-6: A Brief History. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*. https://aclanthology.org/C96-1079

[6] Aric Hagberg, Pieter J. Swart, and Daniel A. Schult. [n. d.]. Exploring network structure, dynamics, and function using NetworkX. ([n. d.]). https://www.osti.gov/biblio/960616

[7] Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *To appear* 7, 1 (2017), 411–420.

[8] Anna Huang, David Milne, Eibe Frank, and Ian H Witten. 2008. Clustering documents with active learning using wikipedia. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 839–844.

[9] Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation Extraction By End-to-end Language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 2370–2381. https://aclanthology.org/2021.findings-emnlp.204

[10] Nishtha Jatana, Sahil Puri, Mehak Ahuja, Ishita Kathuria, and Dishant Gosain. 2012. A survey and comparison of relational and non-relational database. *International Journal of Engineering Research & Technology* 1, 6 (2012), 1–5.

[11] Lung-Hao Lee, Chao-Yi Chen, Liang-Chih Yu, and Yuen-Hsien Tseng. 2022. Overview of the ROCLING 2022 Shared Task for Chinese Healthcare Named Entity Recognition. In *Proceedings of the 34th Conference on Computational Linguistics and Speech Processing (ROCLING 2022)*. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP), Taipei, Taiwan, 363–368. https://aclanthology.org/2022.rocling-1.46

[12] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. (2019).

[13] Stefano Marchesin, Laura Menotti, Fabio Giachelle, Gianmaria Silvello, and Omar Alonso. 2023. Building a Large Gene Expression-Cancer Knowledge Base with Limited Human Annotations. *Database* (2023), 1–19.

[14] Jim Melton and Alan R Simon. 1993. *Understanding the new SQL: a complete guide*. Morgan Kaufmann.

[15] Vincenzo Moscato, Giuseppe Napolano, Marco Postiglione, and Giancarlo Sperlì. 2023. Multi-task learning for few-shot biomedical relation extraction. *Artificial Intelligence Review* (2023), 1–21.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[17] Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. 2020. Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*. ACM, 3125–3132.

[18] Christof Strauch, Ultra-Large Scale Sites, and Walter Kriha. 2011. NoSQL databases. *Lecture Notes, Stuttgart Media University* 20, 24 (2011), 79.

[19] Milena Trajanoska, Riste Stojanov, and Dimitar Trajanov. 2023. Enhancing Knowledge Graph Construction Using Large Language Models. *arXiv preprint arXiv:2305.04676* (2023).

[20] Aleksa Vukotic, Nicki Watt, Tareq Abedrabbo, Dominic Fox, and Jonas Partner. 2015. *Neo4j in action*. Vol. 22. Manning Shelter Island.

[21] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).