DS Journal of Digital Science and Technology
Volume 3 Issue 4, 19-25, Oct - Dec 2024
ISSN: 2583-5416 / https://doi.org/10.59232/DST-V3I4P103

Original Article

Constructing Model for Entity Extraction from Specification Documents of the Database

Giang Ma[1], Nhan Pham[1], Bao Thai[1], Thanh Cao[2], Hai Tran[1*]

[1]*Faculty of Information Technology, Ho Chi Minh University of Education (HCMUE), HCM, Vietnam.*
[2]*Faculty of Information Technology, Saigon University (SGU), HCM, Vietnam.*

[*]haits@hcmue.edu.vn

Received: 05 October 2024; Revised: 06 November 2024; Accepted: 01 December 2024; Published: 24 December 2024;

Abstract - Today, automatically generating Entity-Relationship Diagrams (ERDs) from raw data based on software or system requirements is still predominantly performed manually, incurring significant design costs. Recently, several methods have been proposed to assist users in accomplishing these tasks. However, these methods often rely on rigid rule-based approaches, which cannot be generalized across all scenarios of the exact requirement. Despite having better generalization capabilities than rule-based methods, deep learning-based models need more large-scale labeled datasets. Therefore, this paper recognises the similarity between the NL2ERD and the text-to-SQL problems and proposes an approach to transform existing text-to-SQL datasets into NL2ERD data. Combined with data collected from various Natural Language (NL) types, this approach yields a large-scale NL2ERD dataset. Since NL2ERD can be regarded as a specific task of Information Extraction (IE), we employ this dataset for relation extraction modeling. Experimental results demonstrate that our model achieves high performance.

Keywords – Entity Extraction**,** Entity-Relationship Diagrams (ERDs), Text-to-SQL, NL2ERD.

1. Introduction

Designing a data model involves creating a fundamental data structure for the system under analysis and development. The Entity-Relationship (ER) model is commonly used in the data description analysis step and plays a significant role. An ER model consists of entities, the attributes associated with each entity, and their relationships. Relationships can occur between two entities (entity-entity relationship) or between an entity and an attribute (entity-attribute relationship). Given the considerable challenges associated with manually designing an ER model [1], contemporary methodologies [2-4] typically adopt a two-step approach to automate the generation of ER models from Natural Language (NL) statements, such as software requirements. This approach involves the initial extraction of entities and attributes and identifying their interrelationships.

However, current methods (mostly employing heuristic-based rules) often encounter two major limitations. Firstly, a large number of rules are required to handle synonymous words. For instance, there is a commonly recognized rule that specifies when two nouns are linked by "have" or "has," the noun before "have" is considered an entity, while the nouns after it are seen as attributes. However, many sentences use synonyms for "have," such as "own," "contain," and “possess," which are not addressed by this rule. Furthermore, multiple rules are needed to accommodate different sentence structures, some of which may be similar. For example, a commonly used rule is that consecutive nouns separated by commas or “and” are considered attributes [2, 4]. However, the assertion “Customers are described by name and age” can be rephrased as “Customers not only have a name but also have

an age", which does not align with the rule's parameters. This inconsistency can lead to the inability to apply existing rules to new scenarios effectively.

Deep learning-based NL2ERD tends to perform better for diverse tasks than rule-based methods. However, there are two main reasons for the lack of large-scale annotated data necessary for these models, which previous studies have not addressed. Firstly, previous datasets are characterized by a limited number of data items, and the authors provide no publicly available datasets. Most prior studies assess their proposed methods through case studies. An exception is ER-Converter, which is evaluated using a dataset comprising 30 items, although this dataset is not publicly accessible. Secondly, training a deep learning-based model necessitates data items with comprehensive fine-grained annotations, as token-level annotations can substantially enhance the model's performance. However, the datasets utilized in earlier studies lack annotations indicating which tokens in the utterances correspond to specific entities or attributes.

For these reasons, our study proposes to utilize text-to-SQL datasets and apply algorithms to convert them into NL2ERD datasets [5]. Additionally, we collected and labeled an additional dataset, aiming to enrich the data resource. We hope this dataset will contribute value by supplementing the diverse and rich dataset, thereby improving the performance of deep learning models, specifically NL2ERD models.

As NL2ERD can be viewed as a specialized form of the Information Extraction (IE) problem [6], the REBEL relation extraction model [7] is employed in our study for training based on the previously proposed dataset. The REBEL model extracts various entities and relationships between them based on the input text data. To adapt this dataset to the model, we consider entities and attributes in NL2ERD as entities in IE; meanwhile, entity-attribute and entity-entity relationships in NL2ERD will be treated as two types of relationships in IE. After extraction, entities with their attributes and relationships between entities form a complete Entity-Relationship (ER) schema.

The primary focus of our study is to introduce a novel approach that automatically identifies crucial components in requests, encompassing entities, their attributes, and relationships between entities, thereby facilitating the automatic generation of an ER diagram. This innovative method is poised to revolutionize the field of data modeling and natural language processing.

2. Related work

In recent years, numerous studies have been on extracting entities from database descriptions, with these methods predominantly applying rule-based heuristic approaches. A notable example is a tool for conceptual schema design [8] developed to efficiently and accurately convert data descriptions from natural language into Enhanced Entity-Relationship (EER) models. This method relies on analyzing the structural semantics of natural language into data model concepts.

The analysis algorithm maximizes the utilization of syntax and vocabulary-related information to generate detailed analysis results. These results are refined through rules and heuristics, creating an interactive environment between language information and design knowledge. The tool operates interactively to handle ambiguous, incomplete, or redundant information during conversion. The outcome of this research proposes a creative and convenient approach to transforming requests from natural language into data models, opening up wide potential applications in software development and data management.

Building upon this idea, research directions utilizing natural language as input and applying heuristic techniques have been developed and improved in various ways [9-11, 4]. These methods apply semantic heuristics to identify the ER model's corresponding entities, attributes, and relationships. Gomez et al. observed that using a syntactic heuristic combined with knowledge representation yielded feasible and accurate results in identifying

essential factors in the ER model [9]. Another method proposed using Natural Language Processing (NLP) [11] combined with concept graphs to automatically generate conceptual schema models from text descriptions using the Spanish language presents challenges related to complex language constructs, such as noun-verb combinations in Spanish texts. This research aims to establish a comprehensive software production process by analyzing system requirements in an information system. Btoush and Hammad still employ rule-based methods to generate ER models [3], but with several enhancements: using more explicit rules and parsing sentence structures into syntactic trees for more precise rule application, leading to a complete ER diagram.

However, common drawbacks persist, such as the requirement for input to adhere to predefined rules and limiting user customization. Hettiarachchi et al. apply rule-based methods to derive ER models from natural language input and use natural language to transform into SQL commands based on standard rules between ER and SQL [4], such as entity-table, attribute-column, and relationship-foreign key correspondences. Ahmed et al. utilize NLP techniques like Tokenization [2], POS Tagging for data preprocessing and apply rule-based methods to produce ER diagrams represented as schemas. While these methods all employ heuristic approaches with various rules to yield optimal results, the rules are rigid. They may not be widely applicable across natural language description styles, yet they convey meaning effectively.

Based on preprocessing natural language input [12], applied machine learning models, such as Random Forest, Naive Bayes, Decision Table, and SMO, are used to classify entities, attributes, and sub-entities. Compared to previous studies, the application of these models yielded feasible results. However, the software's business requirements proved limiting factors for this method, as it could not extract relationships between entities. Li et al. [5] developed an algorithm to convert data from input in the form of <NL utterance, SQL query> to <NL utterance, Entity, Attribute, Relation> using the similarity between ERD and database schema [13], where tables represent entities, columns represent attributes of those entities, and foreign keys represent relationships between entities. The Spider dataset [14] was utilized, and they transformed this dataset into the NL2ERM dataset, applying it to two information extraction models [15] and REBEL [7].

3. Proposed Method

In Figure 1, the proposed approach describes how relationships and entities can be extracted from the input specification text. In this study, since NL2ERM is regarded as a specialized information extraction problem [6], we utilized the REBEL relation extraction model [7] for training based on the proposed dataset. The REBEL model is employed to extract various types of entities and relationships between them based on the input text, and the output format is structured as follows: ¡triplet¿ subject [¡subj¿ object ¡obj¿ relationship], where each output triplet always consists of one subject and includes multiple relationships with different objects.

To apply the specific dataset to this proposed model, entities and attributes in NL2ERM are considered as entities in Information Extraction (IE) with two classes, “ENTITY” and ”ATTRIBUTE”; meanwhile, the relationships between entities and attributes, and between entities in NL2ERM, are viewed as two types of relationships in IE with ”attribute of” and ”relation with”. After feeding into REBEL to extract relationships, we will generalize entities, attributes, and relationships based on these extracted relationships and entity types, thereby completing the ERD schema.

3.1. Sentence Segmentation

In this step, morphological analysis is applied to the input description. Users input the requested specifications into the provided workspace area. Sentence boundaries are determined, and the text is divided into sentences for analysis. Typically, periods are used to indicate the end of a sentence. All non-word tokens, such as punctuation marks, are removed, plural suffixes in nouns, such as s, es, or ies, are eliminated, and plural entity names are converted to singular forms.

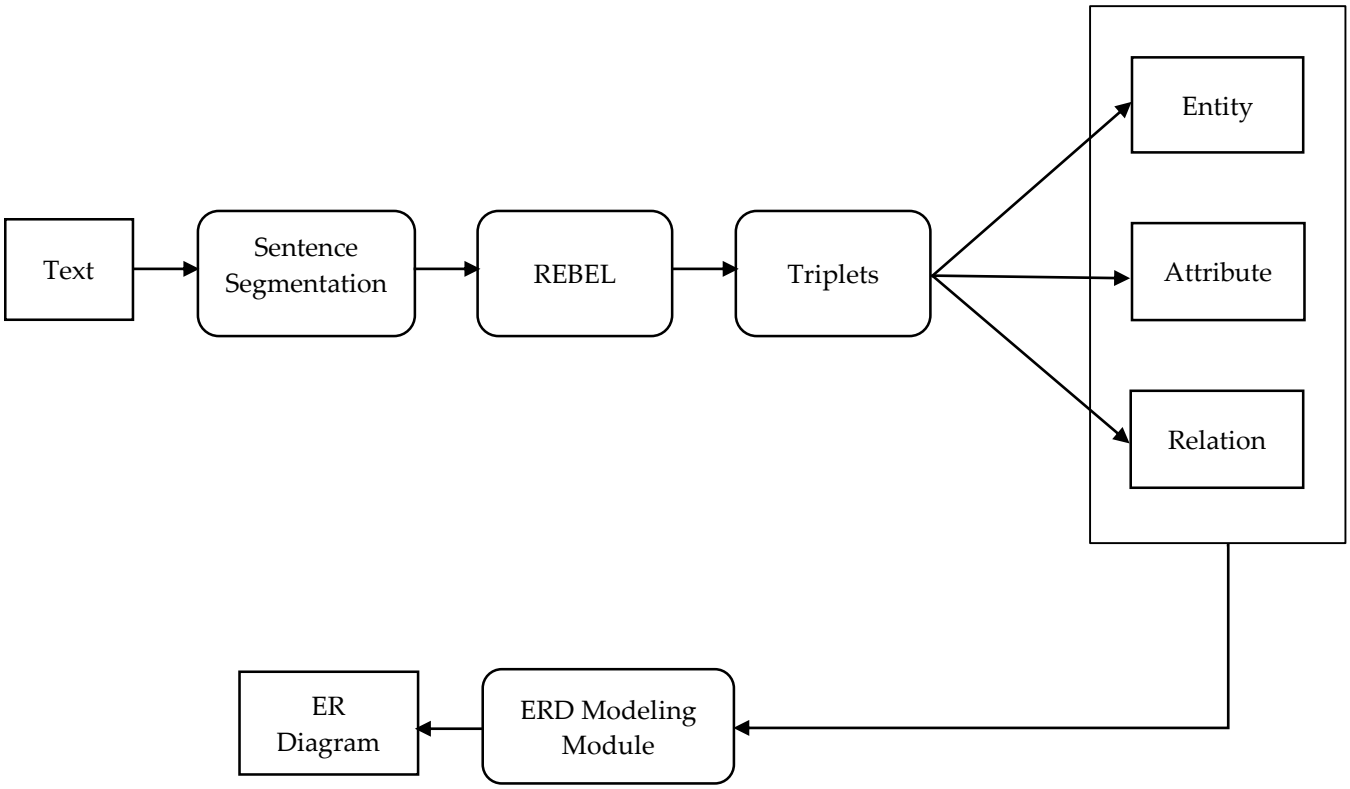

Fig. 1 Process from a raw text to the final ER diagram

3.2. REBEL

REBEL is an advanced method in Extract and Relationship Classification, where the model is considered from a Generative Task perspective. This method is developed based on the BART large language model. REBEL employs a recursive approach to generate descriptions of relationships between entities in text automatically.

During operation, REBEL takes input as text containing entities and implicit relationships between them. This method aims to generate a sequence of entity triples and corresponding relationships that the recursive model can predict and produce. To achieve this, REBEL utilizes linearization to transform entity triples into a token sequence, which the model can more easily generate.

An essential aspect of REBEL is optimizing the number of tokens that need to be decoded. This method employs linearization and specialized techniques to reduce computational costs, increase speed, and ensure accurate results. With this approach, REBEL provides an automatic, efficient, and accurate method for extracting and classifying relationships in text, which can be applied in various fields. Specifically, in our study, it is used to extract relationships between Entities, Attributes, and Relations.

3.3. Triplets

To create a raw entity-relationship diagram, we will rely on the results of the REBEL model. This model's output will be triplets representing the relationships between entities, from which we can extract the entities, their corresponding attributes, and the relationships between entities. These extracted components will then be grouped based on each entity.

3.4. ERD Modeling Model

In this step, we will use the raw entity-relationship diagram from the previous step to visually represent the diagram as a drawing, resulting in a complete entity-relationship diagram.

4. Dataset

In this study, we utilize a combination of two datasets: one from the research by Li et al. [5] and the other from a manually labeled dataset we collected.

4.1. Dataset Based on Transformation Algorithm

The primary data components in this study include a database and a set of natural language sentence pairs linked with schema linking. This linking process refers to identifying entities and attributes in the natural language descriptions and associating them with corresponding elements in the database. Specifically, this schema linking includes an entity and attribute list stored with indexes corresponding to the words in the natural language sentences. In each list, the indexes correspond to words in the natural language sentences identified as entities or attributes. Subsequently, the corresponding entities and attributes in the database descriptions are located using these indexes.

We then utilize the entity and attribute lists in the schema linking, cross-referencing them with the database and the natural language sentence pairs to extract relationships between entities and attributes and between entities. From approximately 14,000 sentence pairs, we refined the dataset to about 10,000 sentences containing the relationships above to fit the model requirements of our study.

4.2. Manually Labeled Data

The dataset we collected is manually labeled. Thus, each sentence is used to define a complete ER schema. Therefore, extracting relationships from these sentences is clearer and more comprehensive. This dataset has been collected and labeled with 1,000 sentences to support the training of the REBEL model.

Table 1. Distribution of sentences in different data types and classes

Data Type	Class	Number of Sentences
Data from Previous Research	Only “attribute of”	10558
	Only “relation with”	2
	Both Classes	106
Manually Labeled Data	Only “attribute of”	660
	Only “relation with”	154
	Both Classes	287
Total		11767

5. Experimental Setup

5.1. Evaluation

We use an Intel i7-9200 processor with a 4M cache of 3.5 GHz for training and testing. After collection, the dataset is cleaned to remove outliers. To evaluate the performance of the proposed model during training, we use three metrics: Precision, Recall, and F1-Score.

Table 2. Main experimental result

	Precision	Recall	F1-Score
attribute of	0.93	0.96	0.94
relation with	0.81	0.75	0.78
All	0.88	0.85	0.86

5.2. Discussion

We observe very high results for the "attribute of" class, with both metrics exceeding 0.9. In contrast, the "relation with" class achieves only moderate performance (around 0.8). This discrepancy is due to data imbalance, as the "attribute of" class tends to be more prevalent than the "relation with" class. Since our test set contains only about 1,000 sentences, even though the metrics for the "relation with" class are high, the model detects only 18 relations per 1,000 sentences, with the following statistics: True Positives (TP): 9, False Positives (FP): 2, False Negatives (FN): 3, and False Positives (FP): 4. However, this disparity is acceptable in practice because an ER schema typically describes a more significant number of entity attributes compared to relationships between entities. This leads to data imbalance and results in only moderate performance metrics for the "relation with" class.

6. Conclusion

In this study, the proposed method aims to extract an ER schema from input text by extracting relationships. The data transformation algorithm and data labeling and training have yielded promising results in our proposed model. Additionally, we collected software requirements, detailed interpretations, and corresponding ER schemas for testing.

However, our research has several limitations. First, relationships in an ER diagram include one-to-one (1-1), one-to-many (1-n), and many-to-many (n-n), while our model groups all these into a single "relation" category. This simplification can result in incomplete ER diagrams, leading to errors when converting to other logical schemas (e.g., many-to-many relationships between two entities may require an additional table to represent the relationship when converted to a database schema). Second, the collected and labeled data is limited. We had to apply the rule: " The attributes must belong to the entity in the same sentence" ([4]; Hettiarachchi et al., 2019) to narrow down the study scope. Without this constraint, additional challenges would arise in linking entities and attributes across multiple sentences. Third, classifying Entity and Attribute as two types of entities in the relationship extraction model leads to frequent misclassification between these entities during testing, indicating that further research is needed for better classification of Entity and Attribute.

References

[1] Nazlia Omar, Paul Hanna, and Paul Mc Kevitt, "Heuristic-Based entity-Relationship Modelling through Natural Language Processing," *Proceeding of the 15th Artificial Intelligence and Cognitive Science Conference*, Artificial Intelligence Association of Ireland, 2004. [Google Scholar] [Publisher Link]

[2] Mudassar Adeel Ahmed et al., "A Novel Natural Language Processing Approach to Automatically Visualize Entity-Relationship Model from Initial Software Requirements," *2021 International Conference on Communication Technologies (ComTech)*, Rawalpindi, Pakistan, pp. 39-43, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] Eman S. Btoush, and Mustafa M. Hammad, "Generating ER Diagrams from Requirement Specifications Based on Natural Language Processing," *International Journal of Database Theory and Application*, vol. 8, no. 2, pp. 61-70, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[4] Sashini Hettiarachchi et al., "A Scenario-Based ER Diagram and Query Generation Engine," *2019 4th International Conference on Information Technology Research (ICITR)*, Moratuwa, Sri Lanka, pp. 1-5, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[5] Zhenwen Li, Jian-Guang Lou, and Tao Xie, "Data Transformation to Construct a Dataset for Generating Entity-Relationship Model from Natural Language," *arXiv Preprint*, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] Christina Niklaus et al., "A Survey on Open Information Extraction," *arXiv Preprint*, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[7] Pere-Lluís Huguet Cabot, and Roberto Navigli, "REBEL: Relation Extraction by End-to-End Language Generation," *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 2370-2381, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] A. Min Tjoa, and Linda Berger, "Transformation of Requirement Specifications Expressed in Natural Language into an EER Model," *Entity-Relationship Approach - ER '93*, pp. 206-217, 1993. [CrossRef] [Google Scholar] [Publisher Link]

[9] Fernando Gomez, Carlos Segami, and Carl Delaune, "A System for the Semiautomatic Generation of ER Models from Natural Language Specifications," *Data & Knowledge Engineering*, vol. 29, no. 1, pp. 57-81, 1999. [CrossRef] [Google Scholar] [Publisher Link]

[10] Luisa Mich, "NL-OOPS: From Natural Language to Object Oriented Requirements Using the Natural Language Processing System LOLITA," *Natural Language Engineering*, vol. 2, no. 2, pp. 161-187, 1996. [CrossRef] [Google Scholar] [Publisher Link]

[11] Azucena Montes et al., "Conceptual Model Generation from Requirements Model: A Natural Language Processing Approach," *Natural Language and Information Systems*, pp. 325-326, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[12] P.G.T.H. Kashmira, and Sagara Sumathipala, "Generating Entity Relationship Diagram from Requirement Specification Based on NLP," *2018 3rd International Conference on Information Technology Research (ICITR)*, Moratuwa, Sri Lanka, pp. 1-4, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[13] Dowming Yeh, Yuwen Li, and William Chu, "Extracting Entity-Relationship Diagram from a Table-Based Legacy Database," *Journal of Systems and Software*, vol. 81, no. 5, pp. 764-771, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[14] Tao Yu et al., "Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task," *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3911-3921, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[15] Yaojie Lu et al., "Unified Structure Generation for Universal Information Extraction," *arXiv Preprint*, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] N. Omar, P. Hanna, P. Mc Kevitt, "Semantic Analysis in the Automation of ER Modelling through Natural Language Processing," *2006 International Conference on Computing & Informatics*, Kuala Lumpur, Malaysia, pp. 1-5, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[17] Raghu Ramakrishnan, and Johannes Gehrke, *Database Management Systems*, McGraw-Hill, Inc., 2002. [Google Scholar] [Publisher Link]

[18] Sebastian Riedel, Limin Yao, and Andrew McCallum, "Modeling Relations and their Mentions without Labeled Text," *Machine Learning and Knowledge Discovery in Databases*, pp. 148-163, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[19] Guillaume Lample et al., "Neural Architectures for Named Entity Recognition," *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 260-270, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[20] Mohd Ibrahim, and Rodina Ahmad, "Class Diagram Extraction from Textual Requirements Using Natural Language Processing (NLP) Techniques," *2010 Second International Conference on Computer Research and Development*, Kuala Lumpur, Malaysia, pp. 200-204, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[21] Vinay S., Shridhar Aithal, and Prashanth Desai, "An NLP Based Requirements Analysis Tool," *International Advance Computing Conference*, Patiala, India, pp. 2355-2360, 2009. [Google Scholar]

[22] Frederik Hogenboom, Flavius Frasincar, Uzay Kaymak, "An Overview of Approaches to Extract Information from Natural Language Corpora," *Proceedings of the 10th Dutch-Belgian Information Retrieval Workshop*, pp. 69-70, 2010. [Google Scholar] [Publisher Link]