SEBHA UNIVERSITY JOURNAL OF PURE & APPLIED SCIENCES VOL.20 NO. 1 2021

مجلة جامعة سبها للعلوم البحتة والتطبيقية
Sebha University Journal of Pure & Applied Sciences

Journal homepage: www.sebhau.edu.ly/journal/index.php/jopas

# Performance Evaluation of Supervised Machine Learning Classifiers for Mapping Natural Language Text to Entity Relationship Models

Mussa A. Omar

Department of Computer Science, Faculty of Information Technology, University of Ajdabiya, Libya

**Keywords:**
Entity Relationship Model
Information Extraction
Machine Learning
Machine Learning Classifiers
Natural Language Text

**ABSTRACT**

Transforming natural language requirements into entities involves a thorough study of natural language text. Sometimes mistakes are made by designers when manually performing this transformation. Often, the process is time-consuming and inaccurate. Hence, multiple research studies have been performed to assist inexperienced designers in mapping a natural language text into entities and reducing the time and error that such a method entails. This work is part of those studies. Human intervention is a significant constraint for prior studies. In this paper, machine learning classifiers are used to eliminate human intervention. The system performs well in predicting entities and has achieved 85%, 75% and 80% for recall, precision and the F-score, respectively. The system also performs well in predicting nouns which do not represent entities and has achieved 68%, 79% and 76% for recall, precision and the F-score, respectively. The performance level of the system is the same as other model generation tools found in the literature. The system is distinguished from these tools in using machine learning classifiers as a technique for establishing entities with no human intervention. Furthermore, the study finds that when distinguishing entities from other nouns, logic-based classifiers, perceptron-based classifiers and SVM classifiers perform better than statistical learning classifiers. The decision tree classifier, neural network classifier and SVM classifier all work well. The decision tree is the better because it can provide a decision tree that defines when a noun is an entity and when it is not based on given features; this is not the case with the neural network classifier and SVM classifier.

# تقييم أداء مصنفات تعلم الآلة الخاضعة للإشراف عند اشتقاق الكينونات الازمة لبناء مخطط الكينونة العلاقة من نصوص اللغات الطبيعي

موسى أحمد محمد عمر

قسم علوم الحاسب، كلية تقنية المعلومات، جامعة اجدابيا، ليبيا

**الكلمات المفتاحية:**
مخطط الكينونة العلاقة
أستخراج المعلومات
تعليم الالة
مصنفات تعليم الالة الخاضعة للإشراف
معالجة اللغات الطبيعية

**الملخص**

عملية أستخراج الكينونات من النص الذى يصف طريقة عمل نظام ما لغرض بناء مخطط الكينونة العلاقة عملية صعبة و تحتاج الى محلل للنظام لكى يقوم بتحليل النص و فهمه و من ثم أشتقاق الكينونات منه. غالبا ما يرتكب محللو النظام أخطاء عند أستخراج كينونات النظام. غالبا ما تستغرق العملية وقتا طويلا ونتائجها غير دقيقة. لهذا السبب تم اجراء الكثير من البحوث لغرض تسخير الحاسب الألي و ذلك بإستخدام بعض تقنيات الذكاء الاصطناعي و منها معالجة اللغات الطبيعية (Natural Language Processing) لمساعدة محللو النظام ذوى الخبرة المحدودة فى تحليل النص الذى يصف طريقة عمل النظام و أستخراج الكينونات منه مما يسهم فى التقليل من الأخطاء التى تحدث أثناء هذه العملية و الحفاظ على الوقت الذى تحتاج إليه هذه العملية. هذا البحث يعد أمتدادا للألبحاث السابقة فى هذا المجال. غير أن النتائج التى توصلت إليها هذه الأبحاث لإتمام عملية

*Corresponding author:
E-mail addresses: mussa.omar@uoa.edu.ly
*Article History : Received 15 November 2020 - Received in revised form 05 January 2021 - Accepted 06 January 2021*

أستخراج الكينونات لا زالت تحتاج الى تدخل جزئى من الانسان فى الحالات التى لا تستطيع برامج معالجة اللغات الطبيعية معالجتها و تم إنتاج برامج تسمى Semi-automated Software لإتمام العملية المشار إليها. التدخل الجزئى للإنسان لأتمام عملية أستخراج الكينونات يعد العقبة الرئيسية امام النتائج التى توصلت إليها الأبحاث السابقة فى هذا المجال. فى هذا البحث تم أستخدام تقنيات تعليم الألة لغرض التخلص من التدخل البشري فى هذه العملية. إن النظام الذي تم إنتاجه يمكنه أستخراج الكينونات بأستخدام تقنيات تعليم الألة بدون التدخل البشري بمعدل يصل الى 80%. إن النظام الذى تم بناءه له القدرة على التعرف إلى الأسماء التى ليست كينونات بنسبة تصل الى 76%. إن معدل الاداء للنظام الذى تم التوصل أليه مطابق لمعدلات الاداء التى تم التوصل إليها خلال بعض الانظمة السابقة. إن النظام الحالي يتميز عن الانظمة السابقة من حيث أستخدامه تقنيات الالة و من حيث أن النظام يعمل بدون التدخل البشري. و من النتائج التى توصل إليها البحث أن الخوارزميات المستندة إلى المنطق والخوارزميات المستندة إلى الإدراك الحسي وخوارزميات ال Support Vector Machine تعمل بشكل أفضل مقارنة بخوارزميات التعلم الإحصائي عند التمييز بين أسماء الكيانات من الأسماء الأخرى. و يعد مصنف شجرة اتخاذ القرار هو الأفضل من بين جميع المصنفات ذلك لان المصنف يستنتج شجرة لأتخاذ القرار يمكن من خلالها معرفة متى يكون الاسم كينونة و متى لا يكون بناء على خصائص محددة.

## Introduction

When a database is produced, a system must be analysed. System analysis involves four significant phases: the study phase, analysis phase, design phase and implementation phase. These are time-consuming. The phases require efforts of a system analyst. The system analyst uses his knowledge and work experience to complete the phases. Establishing an Entity Relationship Model (ERM) out of natural language text is a significant move that cannot be ignored when constructing a database. Designers in general, and inexperienced designers in particular, face difficulties in attempting to build ERMs as they are not skilled enough to do the job correctly. Problems with the formation of ERMs are set out in [1]. The natural language text used to define a context is a problem in itself as it includes issues such as noise, silence, over-specification, inconsistency, forward reference, wishful thinking and uncertainty. Therefore, a variety of research studies have been undertaken to consider the process. Examples of these are given in [2-25]. There are also several approaches used to map natural language text to ERMs such as case-based approach, linguistics-based approach, ontology-based approach, Pattern-based approach and hybrid approach [1]. Creating semi-automated models is the critical drawback of earlier approaches. Three elements must be extracted for constructing an ERM. The elements are entities, entity attributes, and relationships. The identification of entities is a significant task that must be carried out thoroughly during the development of an ERM. This work supports this mission. The papers' theoretical contribution is to investigate which classifier can do the job of extracting entities properly. Another area of inquiry is whether classifiers can function the same as each other. This research tries to find answers to these questions. The paper is divided into four sections. The second section describes the related work. The pre-processing phase is defined in the third part. The experiment and the outcome of the research are in the fourth section. The fifth section comprises a conclusion and upcoming work.

## Approaches for Mapping Natural Language Text into an ERM

1 Linguistics-based Approach

Chen, in 1976, suggested rules that could help in converting natural language text into an ERM [2]. Some researchers have used Chens' rules to design semi-automated models that can extract an ERM out of natural language texts. The models that rely on the linguistics approach use Chen's rules and human intervention to extract the ERM items from natural language texts. The linguistics-based approach is domain-independent, but it is disabled to solve natural language problems, such as noise, silence, over-specification, contradiction, ambiguity, forward reference and wishful thinking [3]. Examples of the tools that are used in this approach are in [4-16].

2 Ontology-based Approach

In computer science, an ontology is the description of a specific domain. The ontology includes domain entities, entity properties and entity relationships. Using such a description when extracting entities from a natural language text helps to decrease ambiguity and human intervention. However, building a domain-independent ontology is problematic and time-consuming. Ontology Management and Database Design Environment (OMDDE) [17] and DC-Builder [18] are examples of the tools that are used in an ontology-based approach to extract entities from natural language texts.

3 Multiple Approaches

The purpose of this approach is to use more than one approach to design a model that can extract entities from natural language texts. The linguistic approach is domain-independent, but it cannot solve natural language problems. Combining the linguistic and ontology-based approaches can produce a model which performs better than if the models are used individually. The Entity Instance Pattern WordNet (EIPW) [19] and Heuristic Based Technique (HBT) [20] use multiple approaches to extract the ERM from natural language texts.

4 Machine Learning Approach

Omar and Abdulla [25] used a machine learning classifier to retrieve entities from a natural language text. The following is the knowledge contribution obtained from the approach:
1. A machine learning approach can deduce entities from natural language texts for conceptual models.
2. A dataset of 1,000 records was produced and used by classifiers in machine learning to distinguish noun entities from others.
3. A fully automated system which extracts entities from natural language texts without human involvement can be produced.
The approach uses appropriate linguistic features for obtaining the candidate list of entities within a natural language text. The machine learning classifier is then used to identify the entities. The system is fully automated and up to 85% accuracy was achieved. More examples of the tools that are used in this approach are in [30-32].

## Preprocessing Stage

This section covers how data is collected, how missing and categorical data are handled, features scaling, handling an imbalanced dataset and splitting data. The dataset which is used in this research, is presented in [25]. Although there are many datasets used for machine learning purposes such as Kaggle Dataset and many others, the author was not succeed in finding a suitable dataset for this experiment. Alternatively, the author looked at the literature. Omar and Abdulla [25] produced a dataset for training a Naive Bayes classifier to differ noun entities from other nouns. The difference is based on nouns features such as common nouns, sentence subject, sentence object, strong entities and noun frequency. There are several parallels between what Omar and Abdulla achieved and what this study wanted the author to accomplish. This is what made the author use the dataset used by Omar and Abdulla for this analysis. Table 1 represents part of the dataset.

**Table 1: Dataset Portion**

| Common Noun | Sentence Subject | Sentence Object | Strong Entity | Frequency |
|---|---|---|---|---|
| Yes | No | No | Yes | No |
| Yes | No | No | Yes | Yes |
| Yes | No | No | Yes | Yes |
| Yes | Yes | No | Yes | Yes |
| Yes | No | No | Yes | Yes |
| Yes | No | No | Yes | Yes |
| Yes | No | No | Yes | No |
| No | No | No | No | No |
| Yes | No | No | No | Yes |
| No | No | No | No | No |

The dataset contains a thousand instances. In 1976, Chen was the founder of the ERD [2]. In 1983, Chen proposed rules to map the text of natural language into an ERD [21]. Chen rules are used as a guide for all the studies that attempted to map natural language text into the ERDs. The studies carried out by [7, 15, 21-22] are an extension to Chen's rules. As a guide, the author selected standard rules in [7, 15, 21-22] to pick a set of characteristics that distinguish entities from other nouns. Common nouns, sentence subjects, sentence objects and strong entities represent entities [7], [21]. Also, the high frequency of a noun is a sign that it might be an entity. Within the dataset, there are no missing values. Therefore, there is no need to handle missing data. However, the dataset contains categorical data which are non-numerical and, thus, need to be converted so that the classifiers can process them. For example, the common noun feature has two categories which are Yes and No. This is the same with the other features. There are many techniques to encode categorical variables for modelling, the two most common of which are Integer Encoding and One Hot Encoding. Integer Encoding means each unique label is mapped onto an integer. Table 2 represents a part of the dataset encoded using this strategy.

**Table 2: A Portion of the Dataset Encoded Using Integer Encoding Strategy**

| Common Noun | Sentence Subject | Sentence Object | Strong Entity | Frequency |
|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 |

One Hot Encoding is a technique to make the categorical variables into a series of dichotomous variables (variables that can have a value of zero or one only). For all but one of the levels of the categorical variables, a new variable will be created that has a value of one for each observation at that level and zero for all others. Table 3 shows a part of the dataset encoded using One Hot Encoding.

**Table 3: A Part of the Encoded Dataset Using One Hot Encoding**

| CN | | SY | | SO | | SE | | F | | E | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Y | N | Y | N | Y | N | Y | N | Y | N | Y | N |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

Table Keys:
CN: Common Noun, SY: Sentence Subject
SO: Sentence Object, SE: Strong Entity,
F: Frequency
Y: Yes, N: No

Using the One Hot Encoding strategy involves removing the last column of each feature. No column was removed because it is the last column of each feature. As a result, Table 4 is an update of Table 3.

**Table 4: A Part of the Dataset Encoded Using One Hot Encoding**

| CN | SS | SO | SE | Frequency | Entity |
|---|---|---|---|---|---|
| Y | Y | Y | Y | Y | Y |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |

Table Keys:
CN: Common Noun, SY: Sentence Subject
SO: Sentence Object, Y:Yes

A comparison is made between Table 2, which represents the dataset encoded using the Integer Encoding Strategy, and Table 4, which represents the dataset that was encoded using One Hot Encoding. Regardless of the coding strategy used, the overall effect of the categorical variable will remain the same. In this experiment, a basic strategy is used for encoding the categorical data of the dataset. There are five features within the dataset. It is crucial to ensure that all of these features have an impact on classifying the nouns into entities. Backward Elimination, Forward Elimination and Bidirectional Elimination are statistical methods used for dimensionality reduction and for eliminating needless features. The methods are applied to the dataset. As a result, the common noun feature and sentence object have been removed from the dataset. Table 5 represents a part of the dataset after elimination of the common noun feature and sentence object feature.

**Table 5: Represents a Portion of the Dataset after Removal of Unnecessary Features Using Backward, Forward and Bidirectional Elimination**

| Sentence Subject | Strong Entity | Frequency- | Entity- |
|---|---|---|---|
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 |

The dataset includes 826 instances in the training set categorised as non-entities and only 174 instances of entities representing nouns.

This is confirmation that the dataset is imbalanced. The imbalanced dataset was converted into a balanced dataset using the Synthetic Minority Over-sampling Technique (SMOTE). Using SMOTE techniques increased the instances which represented the minority class up to 870. The size of the dataset was updated to 1,696. The dataset was divided into a training set and a test set: 80% of the dataset was used for training the classifiers, and 20% was used for testing the classifiers.

**Experiment and Result Discussion**

The experiment deliberated how machine learning classifiers help in mapping nouns onto entities in natural language texts.
Machine learning strategies that incorporate of artificial intelligence systems aim to derive patterns learned from historical data [26]. Kotsiantis et al. [27] and Sen et l. [28] divided machine learning classifiers into four groups: logic-based algorithms, perceptron-based algorithms, statistical learning algorithms and support vector machine-based algorithms. In this paper, the author sought to find out to what extent former algorithms work on mapping nouns onto entities in natural language texts. Do former algorithms work the same way as they do with each other? Is one group better than another when separating entities from nouns? Five classifiers were selected by the authors to evaluate this proposal. The classifiers chosen were a decision tree, neural network, SVM, Naïve Bayes classifier and the ensemble voting classifier. The decision tree classifier represented the logic-based algorithms. The neural network classifier emulated the perceptron-based algorithms. An example of statistical learning algorithms was the Naïve Bayes classifier. The SVM classifier was an algorithm for the SVM-based algorithms. The classifiers were trained on the training set. Table 6 illustrates part of the actual answers and classifier predictions.

**Table 6: Part of Actual Answers and Classifier Predictions**

| Actual Answers | Prediction Answers | | | | |
|---|---|---|---|---|---|
| | DT | NN | SVM | NB | EV |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |

Table Keys:
DT: Decision Tree
NN: Neural Network
SVM: Support Vector Machine
NB: Naïve Bayes
EV: Ensemble Voting

The testing dataset was tested using the former classifiers and assembled to predict the final output. The final output for the ensemble classifier was taken by a majority vote of the classifiers, as shown in Table 6 column 6. Table 7 shows the classifiers outcome predictions.

**Table 7: Outcome Prediction for Classifiers**

| Classifier Name | Class | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Decision Tree | 0 | 0.79 | 0.68 | 0.73 |
| | 1 | 0.75 | 0.85 | 0.80 |
| Neural Network | 0 | 0.80 | 0.66 | 0.72 |
| | 1 | 0.75 | 0.85 | 0.79 |
| Support Vector Machine | 0 | 0.79 | 0.68 | 0.73 |
| | 1 | 0.75 | 0.85 | 0.80 |
| Naïve Bayes | 0 | 0.67 | 0.80 | 0.73 |
| | 1 | 0.79 | 0.66 | 0.72 |
| Ensemble Voting | 0 | 0.79 | 0.68 | 0.73 |
| | 1 | 0.75 | 0.85 | 0.80 |

From Table 7, it can be seen that the system is capable of defining entities with scores of 85% for recall, 75% for precision and 80% for the F-score. The system is capable of defining nouns that do not represent entities with a score of 68% for recall, 79% for precision and 76% for the F-score. Logic-based algorithms, perceptron-based algorithms and SVM algorithms work better as group classifier than statistical learning algorithms when distinguishing entities nouns from other nouns. The decision tree, neural network classifier and SVM classifier all work well in such task. The decision tree is the best because it can give a decision tree that explains when a noun is an entity and when it is not based on any given features; this is not the case for the neural network classifier or the SVM classifier. Table 8 shows a comparison between our system and other model generation tools found in the literature. The comparison based on tool names, year of creating the tool, used technique and limitation.

**Table 8: A Comparison between the System and Existing Model Generation Tools**

| Tool Name | Year | Used techniques | Limitation |
|---|---|---|---|
| CM-Builder [8] | 2003 | Heuristics and NLP | Human intervention |
| ER-Converter [12] | 2004 | Heuristics | Human intervention |
| ACDM [7] | 2008 | Heuristics and typed dependency | Human intervention |
| DBDT [29] | 2009 | Controlled language | Controlled languages |
| Class-GEN [24] | 2011 | Heuristics and NLP | Human intervention |
| Our system | 2020 | Machine learning Classifiers | Fully automated no human intervention |

The author looked at previous studies which map the text of natural language into ERMs. See [7-8, 12, 24, 29] for some of these reports. The author also tested the level of these tools' output. Although the datasets used for testing the tools were different, the output level was between 70-85% using metrics such as Recall and Precision. The critical drawback of the studies is human involvement. To the best of the author's knowledge, Only systems used machine learning classifiers as a tool for mapping natural language text into ERMs are the proposed system and system produced by Omar and Abdulla [25]. Human interaction was discarded, and a fully automated system was developed when machine learning classifiers used.

**Conclusion and Future Work**

Novice designers fail to deduce ERMs from natural language text. Such designers also face difficulties in identifying entities that define a problem domain in a natural language text. Therefore, several analytical studies have been carried out to promote the extraction of entities for inexperienced designers. The critical drawback in recent research has been human involvement. In this research, machine learning classifiers were used to dispense with human involvement in the process. The classifier decision tree is the best classifier that can accomplish such task. The system performs well in predicting entities and achieved 85%, 75% and 80% scores for recall, precision and the F-score, respectively. The system is also successful when predicting nouns which do not represent entities and achieved 68%, 79% and 76% scores for recall, precision and the F-score, respectively. The performance level of the system is the same as other model generation tools found in the literature. The system is distinguished from the existing model generation tools in using machine learning classifiers as a technique for finding entities without human intervention. The system is useful in assisting inexperienced designers in defining entities as the initial step in ERM construction. The authors are interested in exploring the degree to which reinforcement learning decreases human interference and promotes the process of translating natural language texts describing a problem domain into ERMs. This represents a significant research direction and potential for future research.

**References**

[1]- Song. I.-Y., Zhu. Y., Ceong, H., & Thonggoom, O, (2015). “Methodologies for Semiautomated Conceptual Data Modelling

from Requirements,”. In 34th International Conference on Conceptual Modelling, Stockholm, Sweden, 18-31.

[2]- Chen, P. P. S. (1976). “The entity-relationship model-toward a unified view of data,” ACM Trans. Database Syst., 1(1), 9-36. doi: 10.1145/320434.320440.

[3]- Meyer, B. (1985). “On formalism in specifications,” IEEE Software, 1(2), 6-26.

[4]- Gomez, F., Segami, C., & Delaune, C. (1999). “A system for the semiautomatic generation of E-R models from natural language specifications,”. Data & Knowledge Engineering, 29(1), 5781.doi: https://doi.org/10.1016/S0169-023X(98)00032-9.

[5]- Buchholz, E., Cyriaks, H., Düsterhöft, A., Mehlan, H., & Thalheim, B. (1995). “Applying a natural language dialogue tool for designing databases,”. In Proceedings of the First International Workshop on Applications of Natural Language to Databases (NLDB), Versailles, France, 119-133.

[6]- Burg, J., & van de Riet, R. (1998). “Color-x: Using knowledge from wordnet for conceptual modelling,”. In C. Fellbaum & G. Miller (Eds.). “WordNet, An Electronic Lexical Database,” Cambridge, MA: MIT Press, 353-377.

[7]- Du, S. (2008). “On the use of natural language processing for automated conceptual data modelling (PhD thesis),” University of Pittsburgh. Retrieved from http://dscholarship.pitt.edu/8965/1/du-siqing.pdf.

[8]- Harmain, H. M., & Gaizauskas, R. (2003). “CM-Builder: A Natural Language-Based CASE Tool for Object-Oriented Analysis,” Automated Software Engineering, 10(2), 157-181. doi:10.1023/A:1022916028950.

[9]- Meziane, F., & Vadera, S. (2004). “Obtaining ER diagrams semi automatically from natural language specifications,” In Sixth International Conference on Enterprise Information Systems (ICEIS 2004). Porto, Portugal, 638-642.

[10]- Kim, Y. M., & Lee, T. H. (2020). Korean clinical entity recognition from diagnosis text using BERT. BMC Medical Informatics and Decision Making, 20(7), 1-9.

[11]- Omar, N., Hanna, J. R. P, & McKevitt, P. (2004). “Heuristic-based entity-relationship modelling through natural language processing,” In Proc. of the 15th Artificial Intelligence and Cognitive Science Conference (AICS), Galway-Mayo Institute of Technology (GMIT), Castlebar, Ireland, 302-313.

[12]- Han, X., & Wang, L. (2020). A Novel Document-Level Relation Extraction Method Based on BERT and Entity Information. IEEE Access.

[13]- Tjoa, A. M., & Berger, L. (1994). “Transformation of requirement specifications expressed in natural language into an EER model,” In R. Elmasri, V. Kouramajian & B. Thalheim (Eds.), “Entity-Relationship Approach — ER '93,” Lecture Notes in Computer Science, 823, Berlin, Heidelberg: Springer, 206-217.

[14]- Tseng, F. S., Chen, A. L., & Yang, W.-P. (1992). “On mapping natural language constructs into relational algebra through ER representation,” Data & Knowledge Engineering, 9(1), 97118.

[15]- Athenikos, S. J., & Song, I. Y. (2013). “CAM: A Conceptual Modelling Framework based on the Analysis of Entity Classes and Association Types,” Journal of Database Management (JDM), 24(4), 51-80.

[16]- Ambriola, V., & Gervasi, V. (2006). “On the systematic analysis of natural language requirements with circe,”Automated Software Engineering, 13(1), 107-167.

[17]- Sugumaran, V., & Storey, V. C. (2002). “Ontologies for conceptual modelling: their creation, use, and management.,” Data & Knowledge Engineering, 42(3), 251-271. doi: https://doi.org/10.1016/S0169-023X(02)00048-4.

[18]- Herchi, H. & Abdessalem, W. B. (2012). “From user requirements to UML class diagram,” In International Conference on Computer Related Knowledge, Sousse, Tunisia. Retrieved from http://arxiv.org/abs/1211.0713.

[19]- Thonggoom, O., Song, I.-Y., & An, Y. (2011). “EIPW: A Knowledge-Based Database Modelling Tool,” In C. Salinesi & O. Pastor (Eds), “Advanced Information Systems Engineering Workshops,” CAiSE 2011. Lecture Notes in Business Information Processing, 83. Berlin, Heidelberg: Springer.

[20]- Thonggoom, O. (2011). “Semi-automatic Conceptual Data Modelling Using Entity and Relationship Instance Repositories (PhD thesis),” Drexel University, Philadelphia, PA, USA.

[21]- Chen, P. P. S. (1983). “English sentence structure and entity-relationship diagrams,” Information Sciences, 29(2), 127-149.

[22]- Hartmann, S., & Link, S. (2007). “English sentence structures and EER modelling,” In Proceedings of the fourth Asia-Pacific conference on conceptual modelling - Volume 67, Ballarat, Australia, 27-35.

[23]- Overmyer, S. P., Lavoie, B., & Rambow, O. (2001). “Conceptual modelling through linguistic analysis using LIDA,” In Proceedings of the 23rd international conference on Software engineering, Eden Roc Renaissance, Miami Beach, USA, 401-410.

[24]- Elbendak, M. E. (2011). “Requirements-driven Automatic Generation of Class Models (PhD thesis),” Northumbria Univeristy, Newcastle upon Tyne.

[25]- Omar, M., Abdulla, A. (2020). “The Entities Extraction for Entity Relationship Models from Natural Language Text via Machine Learning Algorithms,” In Proceedings of the 4th International Conference of Basic Science and Their Applications, Elbeida City, Libya.

[26]- Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. Expert Systems with Applications, 124, 226-251.

[27]- Kotsiantis, S. B., Zaharakis, I. D., & Pintelas, P. E. (2006). “Machine learning: a review of classification and combining techniques,” Artificial Intelligence Review, 26(3), 159-190.

[28]- Sen, P. C., Hajra, M., & Ghosh, M. (2020). Supervised classification algorithms in machine learning: A survey and review. In Emerging Technology in Modelling and Graphics (pp. 99-111). Springer, Singapore.

[29]- Al-Safadi, L. A. (2009). “Natural language processing for conceptual modelling,”. International Journal of Digital Content Technology and its Applications, 3(3), 47-59.

[30]- Suárez-Paniagua, V., Zavala, R. M. R., Segura-Bedmar, I., & Martínez, P. (2019). A two-stage deep learning approach for extracting entities and relationships from medical texts. Journal of biomedical informatics, 99, 103285.

[31]- Zhang, Z., Zhan, S., Zhang, H., & Li, X. (2020). Joint model of entity recognition and relation extraction based on artificial neural network. Journal of Ambient Intelligence and Humanized Computing, 1-9.

[32]- Liu, Jin, Yihe Yang, and Huihua He. "Multi-level semantic representation enhancement network for relationship extraction." Neurocomputing 403 (2020): 282-293.