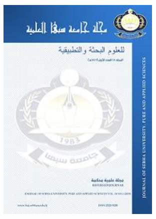
SEBHA UNIVERSITY JOURNAL OF PURE & APPLIED SCIENCES VOL.20 NO. 2 2021
DOI: 10.51984/JOPAS.V20I2.1329

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

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



Novel rules for extracting the entities of entity relationship models

*Mussa A M Omar[1], Abdulrhman A Y Alsheky[2], Balha H N Faiz[3]

[1]Faculty of Information and Technology, Department of Computer Science, University of Ajdabyia, Libya
[2]Faculty of Information and Technology, General Department, University of Ajdabyia, Libya
[3]Faculty of Information and Technology, Information Technology Department, University of Ajdabyia, Libya

Keywords:
Entity relationship models
Natural language processing
Linguistics rules
Machine learning
Logic based classifiers
Mapping nouns into entities

A B S T R A C T

Extracting entities from natural language text to design conceptual models of the entity relationships is not trivial and novice designers and students can find it especially difficult. Researchers have suggested linguistic rules/guidelines for extracting entities from natural language text. Unfortunately, while these guidelines are often correct they can, also, be invalid. There is no rule that is true at all times. This paper suggests novel rules based on the machine learning classifiers, the RIPPER, the PART and the decision trees. Performance comparison was made between the linguistic and the machine learning rules. The results shows that there was a dramatic improvement when machine learning rules were used.

قوانين جديدة لإستخراج كينونات مخطط الكينونة العلاقة

موسى أحمد محمد عمر 1و عبدالرحمن عبدالله مفتاح الشيخى 2 و بلها حسن نصر فايز[3]

[1]كلية تقنية المعلومات ، قسم علوم الحاسب ، جامعة أجدابيا ، ليبيا
[2]كلية تقنية المعلومات ، القسم العام ، جامعة أجدابيا ، ليبيا
[3]كلية تقنية المعلومات ، قسم تقنية المعلومات ، جامعة أجدابيا ، ليبيا

الكلمات المفتاحية:
التعلم الألى
المصنفات المنطقية
تعيين الكينونات من الاسماء
قواعد اللغويات
مخطط الكينونة العلاقة
معالجة اللغات الطبيعية

الملخص

إن أستخراج الكينونات من نص الذى يصف نظام ما لغرض إنشاء مخطط الكينونة العلاقة ليس أمرا سهلاً خاصة للمصممين ذوى الخبرة المحدودة، لذا أقترح الباحثون بعض القواعد الغوية للإستدلال بها أثناء أستخراج الكينونات من النص الذ ى يصف النظام. و نظراً لعدم إمكانية الاعتماد على هذه القواعد كلياً فإن هذه الورقة تقدم قواعد جديدة لإستخراج الكينونات تستند على مصنفات التعلم الآلى بدلاً من القواعد اللغوية. و قد عقد مقارنة بين القواعد اللغوية و القواعد التى تعتمد على مصنفات التعلم الآلى لمعرفة مستوى أختلاف الاداء بينهما، و قد تبين من خلالها التحسن الواضح فى الاداء عند أستخدام مصنفات التعلم الآلى.

Introduction

Creating an Entity Relationship Model (ERM) from requirements is frequently the first step in designing a database system, which is an important step in the software development life cycle. Manually extracting conceptual models is a time-consuming, error-prone task. The use of automated transformation aids in the preservation of requirement traceability [1]. One of the most difficult aspects of developing an automated ERM is the lack of complete natural language rules for mapping natural language requirements into an ERM [2]. The most prevalent means of describing and communicating things is a natural language [3]. Almost 90% of all industrial practices requirements are written in natural language [4], [5]. Limitations exist in transposing a requirement into a conceptual model in natural languages are: the natural language is ambiguous and an effective and accurate analysis is difficult; the same semantics

*Corresponding author:
E-mail addresses: mussa.omar@uoa.edu.ly, (A. A. Y. Alsheky)abdurhman.alsheky@uoa.edu.ly, (B. H . Faiz)balha_h_faiz@uoa.edu.ly

Article History : Received 29 June 2021- Received in revised form 11 August 2021 - Accepted 17August 2021

can be represented in many ways; therefore, the ways in which these correlational semantic differences are managed is necessary. It is consequently necessary to know the realm of hidden entities that have not been expressly declared. Techniques and rules for modelling principles are therefore necessary [3]. There are several approaches for mapping natural language text into a conceptual model, including the Linguistics-based approach, Pattern-based approach, Case-based approach, and Ontology-based approach [6]. These methods are incapable of producing fully automated systems. The authors believe this is due to insufficient natural language rules for mapping natural language requirements into an ERM.

This provided the motivation for our study of the correspondence between the structure of English language sentences and ERMs. In 1983, Chen suggested guidelines/rules for mapping text in natural language into an ERM [7]. However, within requirement specification scripts, the guidelines / rules may not be sufficiently universal to satisfy the syntax variables. Syntax variables are the various ways in which the same object is declared. In addition, the Chen standards can be insufficient, as the guidelines are sometimes correct and sometimes invalid; there is no rule that is true all times. Common nouns may, for instance, represent entities, but not every common noun within a specification of requirements will be mapped into an entity. This raises the question of how common nouns that represent entities can be differentiated from those that do not.

Chen rules are overlapping, and so cannot work together with a broad group of rules. Nouns can, for instance, be mapped into entities, but nouns can also be mapped into attributes. Thus, a difficulty will arise if these two rules are used together as to whether a noun should be mapped into entities or attributes. Therefore, only when creating a method for mapping natural language text into a conceptual model can linguistic rules offer a simple service [6]. Machine learning rule-based classifiers are used to overcome the constraints of existing guidelines / rules that map natural language requirements into ERMs. The purpose of the proposed guidelines is to assist designers in retrieving entities from the text listing requirement specifications.

The contributions to knowledge of this paper are (1) the use of machine learning classifiers in the development of guidelines for mapping natural language text into entities, (2) the development of a set of guidelines for mapping natural language text into entities and (3) a comparison between linguistic guidelines and machine learning guidelines. The remainder of the paper is as follows. In the second section, a summary of similar work is presented. In section three machine learning classifiers were reviewed. In Section four, experimental outcomes are discussed. In Section five, conclusions are presented and a suggestion for future work is made.

Reviewing of Related Work

Linguistic guidelines for extracting entities from natural language text are reviewed in this section.

1. Common nouns are mapped into entities

A common noun in the English language is a word that refers to an item or object, such as a doctor, a school, etc. Entities can be indicated by common nouns [7],[8]. For example, in the sentence “a student registers for a module" the common nouns of student and module are assigned to entities. This rule is not entirely precise as not all common nouns in the text representing the specifications can be fitted to entities.

2. Sentence subjects are mapped into entities

The subject of a sentence is part of the language that describes an event. In this sentence, for example, "students register on programs", "students" represents the subject of the sentence. The subject of a sentence represents an entity [8][9]. However, not all sentence subjects are converted into entities within the requirements of the specification text. In the sentence, "One university includes several schools," although the university is the subject of the sentence it is not an entity because it represents system name. Thus, the subject-matter of the earlier sentence should not be converted into an entity. However, transforming each sentence subject within the requirements of the specification text into an entity can result in incorrect assignment.

3. Sentence objects are mapped into entities

The object of a sentence is the person or thing that is subject to the action of the verb. A phrase object is mapped into an entity [8], [9]. For example, "modules" is a phrase object in the phrase "students work on modules". The object in the previous sentence is transmuted into an entity. The sentence, "a student has date of birth" says, for example, although the date of birth is a sentence object, but it is not an entity. Thus, the sentence object here should not be converted into an entity. Transforming each sentence object within the specification text into an entity can result in incorrect assignment.

4. Strong entities category in WordNet ontology are mapped into entities

WordNet Ontology [10] separates nouns into three categories: strong entities, weak entities, and medium entities. According to WordNet Ontology, a noun is categorized as an entity when its inherited hypernyms includes one of the following categories: group, physical object, physical entity and thing [11]. A string of noun phrase hypernyms is acquired, and if it corresponds to the category of strong entities, then the noun phrase is transformed into an entity. For example, in the sentence “university database records the following information about each student: number, name, address, nationality and birth date”. The noun “name” in the sentence is attributed to student entity. However, according to WordNet Ontology the noun “name” is an entity. The consequence of inherited hypernym for name as a noun is: class, collection, group, abstraction and entity. Thus, it is an entity. Transforming each strong entity within the specification text into an entity can result in incorrect assignment.

5. High frequency nouns are mapped into entities

When a noun has a high frequency of occurrence this is a sign the noun can be mapped into an entity [12]. However, transforming each high frequency noun within the specification text into an entity can result in incorrect assignment.

The rules utilized to map nouns into entities are not comprehensive. There is no single rule that applies in all cases. Common nouns, sentence subjects, and sentence objects can all be assigned to entities, but not every common noun, sentence subject, sentence object, strong entity noun and high frequency noun in a requirement specification text should be assigned to an entity. Fixed and reliable rules for mapping a noun into an entity do not exist. At one time a common noun is mapped into an entity, at other times it is not. This is also applies to sentence subjects, sentence objects, strong entities and high frequency nouns. This apparent disparity is the main motivation for this research. The author hopes to find reliable rules for mapping nouns into entities.

Reviewing of Machine Learning Classifiers

Classifiers for machine learning are reviewed in this section and appropriate classifiers selected to be used to establish guidelines /rules for mapping nouns into entities.

1. Logic based Classifiers

We will focus on two classes of logic (symbolic) learning techniques in this section: decision trees and classifiers based on rules.

Decision trees sort instances and classify them based on feature values. In an instance to be categorized, each node in a decision tree represents an attribute, and each branch represents a value that the node may assume. Starting at the root node, instances are listed and sorted on the basis of their attribute values. A non-deterministic polynomial problem is associated with building optimal binary decision trees, and theoreticians have pursued successful heuristics for the construction of near-optimal decision trees. The root node of the tree will be the feature that best divides the training data. There are many ways to find this feature, such as Information Gain [13] and Gini Index [14]. However, most studies have concluded that there is no single best technique [15]. When determining which metric should be used in a specific dataset, comparison of individual methods can still be useful. On each partition of the divided data, the same process is repeated, generating sub-trees until the training data is divided into subsets of the same class.

For certain definitions, decision trees may be substantially more complex because of the replication problem. To prevent duplication, a solution can use an algorithm to apply complex features on nodes. The C4.5 algorithm for the construction of decision trees is the most well-known [16], and is an expansion of the earlier ID3 algorithm [17]. Recent research compared decision trees and other learning algorithms [18] and found that C4.5 has a very fast with relatively low error. In 2001, Ruggieri provided an empirical review of the C4.5

algorithm's runtime behaviour, which identified some improvements in performance by using a version of the RainForest algorithm, which needed no sorting [19]. Ruggieri introduced a more effective version of the algorithm, called EC4.5, based on his evaluation Ruggieri concluded that his implementation computed the same decision trees as C4.5 but with a gain in performance of up to five times. Comprehensibility is one of the most valuable aspects of decision trees because they allow people to readily understand why an example is categorized as belonging to a particular class.

Decision trees can be transformed to a set of rules by making a separate rule for each route from the root to the tree leaf [16]. However, using a number of rule-based algorithms, rules can also be explicitly induced from training data. Furnkranz has provided an outstanding review of current work using rule-based approaches [20]. RIPPER and PART are two common rule-based algorithms where the aim is to construct the shortest set of rules that are compliant with the training results. A large number of learned rules, rather than exploring the assumptions that govern them is usually an indicator that the learning algorithm is attempting to remember the training set. The distinction among rule learning algorithms and decision tree algorithms is that the latter assesses the overall quality of a variety of disjointed sets, while rule learning algorithms assess only the quality of the set of instances identified by the candidate rule [21].

2. Perceptron based Classifiers

Perceptron can be defined briefly as: if the input function values are x_1 through x_n and the relation weights / prediction vector are w1 through wn then the perceptron calculates the total of the weighted inputs as $\sum I\ x_i * w_i$, and the sum produced is then compared with an adjustable threshold: the sum produced is equated to 1 if the total is greater than the threshold; otherwise it is zero [22]. The algorithms can be based on the perceptron principle.

The most efficient way to learn from batch training of examples is to use the perceptron algorithm and run the algorithm continuously through the training set until it finds a prediction vector that is right for every member of the training set. This prediction rule is then used for identifying the labels on the testing dataset [21].

When dealing with irrelevant attributes, perceptron-like linear algorithms are superior to other algorithms. When there are many attributes this can be a tremendous benefit [23]. Most perceptron-like algorithms do not deal with numerical characteristics, so prior to implementation, numerical characteristics should be discretized [22]. Perceptron-like techniques are binary, and the problem must be reduced to a collection of multiple binary classification problems in the case of a multi-class problem [21].

Perceptrons are only able to distinguish linearly separable instance sets. If the instances are not linearly distinct, learning can never achieve a stage where all instances are properly categorized. To attempt to solve this problem, a class of Artificial Neural Networks (ANNs) has been developed [24]. A multi-layer neural network involves a large number of units (neurons) linked together in a pattern of connections. Net units are commonly divided into three classes: input units, which obtain the information to be processed; output units, where the processing results are found; with hidden units in between. Signals move only in the direction from input to output. In general, it is a challenge to correctly determine the size of the hidden layer, since an underestimation of the number of neurons may lead to weak approximation and generalization capabilities, whereas excessive nodes may lead to overfitting and ultimately make it more difficult to search for the global optimum [21].

ANNs have been adapted to solve many serious problems, but their main notable drawback remains, their failure to reason in a way that efficiently conveys how and why they achieved the given solution. For this reason, several scientists have begun to tackle the issue of enhancing the understandability of neural networks, where extracting symbolic rules from trained neural networks is the desired outcome [22]. Neural networks can typically provide incremental learning more effectively than decision trees [25].

3. Statistical Learning Classifiers

In comparison to ANNs, statistical methods are distinguished by an explicit underlying probability model that provides a probability that a case, rather than merely a classification, belongs in a particular class. Bayesian Networks (BNs) and instance specific approaches are examples of this form of classification algorithm [21].

A BN is a graphical representation of a set of variables (features) and their probability relationships. Compared to decision trees or neural networks, the most important aspect of BNs is, most definitely, the possibility of including prior knowledge on a given topic as intersections of structural relationships between its features. This prior knowledge or domain knowledge of the BN structure will take the following forms: (a) declaring that a node is a root node, i.e. that it does not have a parent, (b) declaring that a node is a leaf node, i.e. that it does not have a sibling, (c) declaring that a node is a direct cause or consequence of another node (d) declaring that a node is not explicitly linked to another node, (e) declaring that two nodes are distinct, provided the condition set, and (f) supplying the ordering of partial nodes, i.e. declaring that the node occurs earlier than the ordering node [16]. An issue with BN classifiers is that they are not sufficient for datasets with several features [26]. The explanation for this is that it is simply not feasible in terms of time and space to create a very large network. A final issue, in most situations, is that the numerical features need to be discretized before induction [22].

Naive Bayesian Networks (NBN) are very simplistic BNs consisting of Directed Acyclic Graphs with only one parent and many children, with a powerful expectation of independence between child nodes. The key value of the NBN classifier is its short training period. Moreover, because the model has the structure of a product, it can be transformed into a simple sum using logarithms, with considerable computational benefit [21]. The standard practice is to discretize during data pre-processing if a feature is numerical [27], but a researcher can use the regular Gaussian distribution to determine probabilities [28].

Instance-based learning algorithms are lazy-learning algorithms, since they delay the process of inference or generalization before classification is carried out. During the training period, lazy-learning algorithms need less computation time than eager-learning algorithms (such as decision trees, NNs and BNs), but more computation time during the process of classification. The nearest neighbor algorithm is one of the simplest instance-based learning algorithms [21].

Perceptron is extensively employed to enable relation extraction because they offer the advantage of automatically extracting high-order representation from relation instances [29]. Perceptron is not suitable for developing guidelines for detecting entities from natural language specification text because the understanding of the internal functioning of neural networks is not sufficiently understood. For generating guidelines for classifying entities from natural language specification text, statistical methods are also not sufficient, since statistical methods, fundamentally, provide a probability that a case, rather than merely a classification, belongs in each class. Logic based classifiers are the most effective classifiers for distinguishing entities from natural language text. One of the most important features of Logic-based classifiers is their understandability. People can readily understand why a decision trees and sets of learning rules categorize an example as belonging to a specific class. The authors utilized decision tree classifiers, the PART and the RIPPER classifiers for developing a set of guidelines to be used for entity identification from natural language text.

4. Experiment and Results Discussion

The purpose of the experiment is use machine learning classifiers for establishing a set of rules to be utilized for detecting entities of ERMs from natural language text. The rules will be compared against linguistic rules found in the literature. The result of the comparison will reflect which set of rules works better in distinguishing nouns that represents entities from other nouns. To train the system, a training set is constructed. This training set consists of a series of twenty-six case studies, obtained from authentic material. All case studies include predefined solutions that provide a list of entities. To define common nouns, sentence subjects, and sentence objects, each case study is first inserted into Stanford CoreNLP*, then the case study is inserted into Rita WordNet† to define strong entities. The training set included a thousand records. Table 1 represents part of the data.

Table 1: Dataset portion.

CN	SS	SO	SE	F	E
Yes	No	No	Yes	No	No
Yes	No	No	Yes	Yes	Yes
Yes	No	No	Yes	Yes	Yes
Yes	Yes	No	Yes	Yes	Yes
Yes	No	No	Yes	Yes	No
Yes	No	No	Yes	Yes	Yes
Yes	No	No	Yes	No	Yes
No	No	No	No	No	No
Yes	No	No	No	Yes	No
No	No	No	No	No	No

Table Keys:
CN: Common Noun
SS: Sentence Subject
SO: Sentence Object
SE: Strong Entity
F: Frequency
E: Entity

The features are selected according to a set of rules given in Section Two. Categorical data was not converted to numerical data because decision tree classifiers, RIPPER and PART, work with categorical data. In the training set, eight hundred and twenty-six instances are listed as non-entities and only one hundred and seventy-four instances represent noun entities. This demonstrates the imbalance of the dataset. By using the Synthetic Minority Over-sampling Technique (SMOTE), the imbalanced dataset was converted into a balanced dataset. With the balanced data there are 1,600 instances. The training set contains 1300 instances while the test set contains 300 instances. The training set was trained using rules classifiers including RIPPER and PART. Table 2 shows obtained rules.

Table 2: Rules generated by ripper and part classifiers.

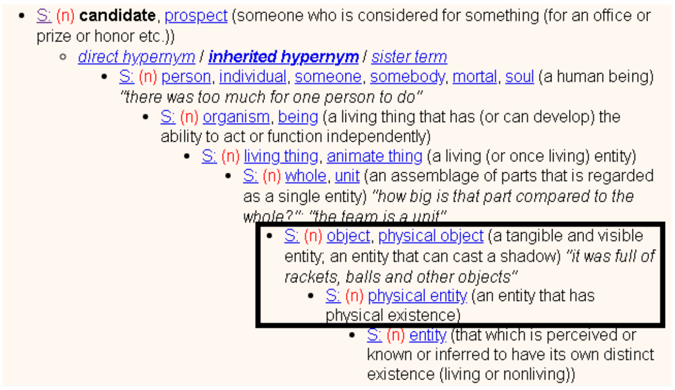
NO	Rule	The classifier
MLR1	Frequency = Yes AND Strong entity = Yes AND Sentence subject = Yes => Entity=Yes	RIPPER
MLR2	Frequency = Yes AND Strong entity = Yes and Sentence object = No => Entity=Yes	RIPPER
MLR3	Common noun = Yes AND Strong entity = Yes AND Sentence object = No=> Entity=Yes	PART
MLR4	Frequency = No => Entity=No	PART
MLR5	Common noun = No => Entity=No	PART
MLR6	Sentence object = Yes AND Sentence subject=No => Entity=No	PART
MLR7	Sentence object = No AND Sentence subject=No => Entity=Yes	PART

Following is a short explanation for above rules:
Rule 1: A noun is an entity when it meets the following criteria:

1. There is frequency to the noun. This means that in the natural language text that provides the problem statement, the noun appears at least twice.
2. The noun belongs to strong entity groups.
3. The noun signifies a sentence subject in the natural language text.

Example 1: "A Temporary Employment Corporation (TEC), places temporary workers in companies. TEC has a file of candidates who are willing to work. Each candidate has one or more qualifications. Each qualification may be held by one or more candidates. If a candidate has worked before, the candidate has a specific job history. Each time the candidate works, an additional job history record is created" [11, p. 132].

The frequency of occurrence of the common noun "candidate" is six. WordNet Ontology is utilized for finding inherited hypernyms of "candidate" as given in Fig. 1.



- S: (n) **candidate**, prospect (someone who is considered for something (for an office or prize or honor etc.))
  - direct hypernym / ***inherited hypernym*** / sister term
    - S: (n) person, individual, someone, somebody, mortal, soul (a human being) *"there was too much for one person to do"*
      - S: (n) organism, being (a living thing that has (or can develop) the ability to act or function independently)
        - S: (n) living thing, animate thing (a living (or once living) entity)
          - S: (n) whole, unit (an assemblage of parts that is regarded as a single entity) *"how big is that part compared to the whole?"; "the team is a unit"*
            - S: (n) object, physical object (a tangible and visible entity; an entity that can cast a shadow) *"it was full of rackets, balls and other objects"*
              - S: (n) physical entity (an entity that has physical existence)
                - S: (n) entity (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

Fig 1: Inherited hypernyms of candidate as a noun

If an inherited hypernym shows the noun is an object / a physical object, then the noun is a strong entity. A noun is classified as an entity when one of the following categories is included in its inherited hypernyms: group, physical object, physical entity and thing [11]. Fig. 2 shows Stanford Dependency for the 3rd sentence in Example 1.

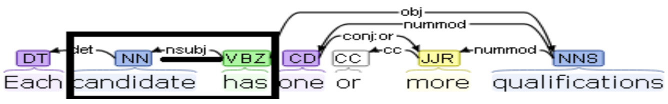
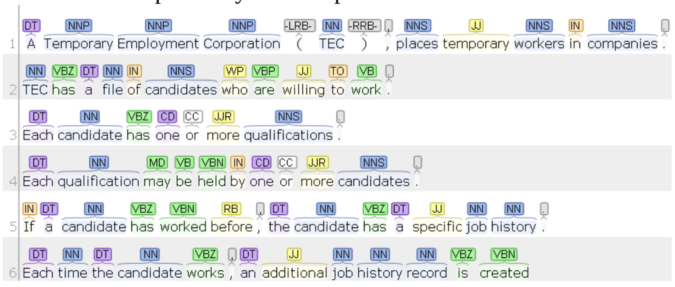


Fig 2: Stanford dependency of 3rd sentence in example 1

Fig. 2 demonstrates that the candidate noun is a sentence subject of the third sentence. The candidate has fulfilled all the criteria required in Rule 1, consequently it is an entity.

Rule 2: A noun represents an entity when it meets following requirements:
1. The noun appears at least twice in the natural language text that provides the problem statement.
2. The noun belongs to groups of strong entities.
3. The nouns do not appear in the natural language text as a sentence object.

In Example 1, the frequency of the common noun "candidate" is six. Fig. 1 confirms that the noun belongs to the object, a physical object and physical entity, which are strong entity groups. Fig. 3 illustrates the Stanford Dependency of Example 1.


DT NNP NNP NNP -LRB- NN -RRB- , NNS JJ NNS IN NNS .
NN VBZ DT NN IN NNS WP VBP JJ TO VB .
DT NN VBZ CD CC JJR NNS .
DT NN MD VB VBN IN CD CC JJR NNS .
IN DT NN VBZ VBN RB , DT NN VBZ DT JJ NN NN .
DT NN DT NN VBZ , DT JJ NN NN NN VBZ VBN

1 A Temporary Employment Corporation (TEC) , places temporary workers in companies .
2 TEC has a file of candidates who are willing to work .
3 Each candidate has one or more qualifications .
4 Each qualification may be held by one or more candidates .
5 If a candidate has worked before , the candidate has a specific job history .
6 Each time the candidate works , an additional job history record is created

Fig 3: Shows Stanford Dependency of Example 1

Although the noun has been listed six times within the problem statement, the sentence object does not reflect any of these. In other words, the noun meets the criteria of Rule 2, and is an entity.

Rule 3: A noun indicates an entity when the following conditions are met:
1. The noun is a common noun.
2. The noun belongs to strong groups of entities.
3. The nouns do not appear as a sentence object in the natural language text that defines the problem statement.

* https://corenlp.run
† http://www.rednoise.org/rita/index.html

Example 2: “Each candidate has one or more qualifications.”
When the POS , tagger which is part of Stanford CoreNLP, is utilized for assigning each token type within a stated sentence of Example 2, a NN tag is assigned as part of a speech tagger for the noun candidate, as illustrated in Fig. 4.

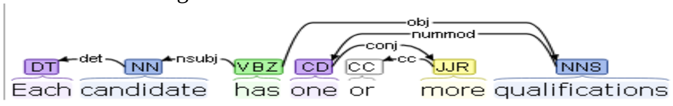

DT det NN nsubj VBZ CD CC JJR NNS
obj
nummod
conj
cc
Each candidate has one or more qualifications

Fig 4: POS tagger for stated sentence of example 2

A NN tag means the token is classified as a common noun. The noun which is the candidate in example number two is a common noun. Fig. 1 confirms the noun belongs to an object, a physical object and a physical entity, which are strong entity groups. Fig. 5 shows Stanford Dependency of Example 2.

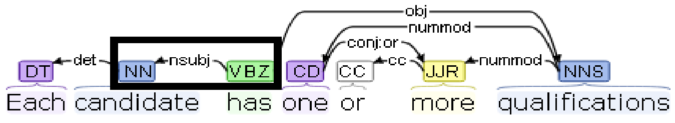

DT det NN nsubj VBZ CD CC JJR nummod NNS
obj
nummod
conj:or
cc
Each candidate has one or more qualifications

Fig 5: Shows Stanford Dependency of example 2

Fig. 5 shows the candidate noun plays the role of the subject in the sentence, while the qualification represents the role of the object of the sentence. In Example 2, the candidate noun is a common noun; the noun belongs to the objects, physical objects and physical entities, which are strong classes of entities and do not represent the position of the object in the sentence. It is then mapped into an entity.
Rule 4: A noun does not represent an entity if it does not have frequency within a natural language. A common noun will not be considered as an entity, unless it has a frequency of more than one.
Under Rule 4, the common noun “candidate” defined in Example 2 is not mapped into an entity unless it is referred to more than once in the text of the problem statement.
Rule 5: A noun, unless it is a common noun, does not represent an entity. While there are other features that support a noun being mapped into an entity, such as being the subject or object of the sentence, and frequency, a noun is not mapped into an entity unless it is a common noun.
Rule 6: If the following conditions are met, a noun is not mapped into an entity.
1. The noun performs the role of the sentence object within the text of the problem statement.
2. The noun does not play the role of representing the subject of a sentence within the text of the problem statement.
Example 3:“Each candidate has one or more qualifications. If a candidate has worked before, the candidate has a specific job history. Each time the candidate works, an additional job history record is created.” [11, p. 132]. Fig. 6 shows Stanford Dependency of Example 3.

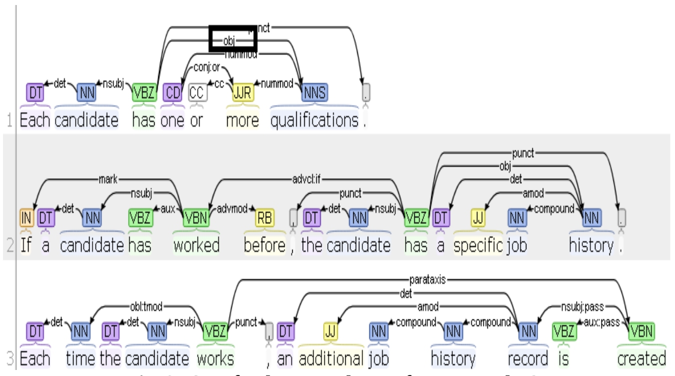

1 Each candidate has one or more qualifications .
2 If a candidate has worked before , the candidate has a specific job history .
3 Each time the candidate works , an additional job history record is created

Fig 6: Stanford Dependency for Example 3

The above text is a requirements specification for a Temporary Employment Corporation. “Qualification” is the sentence object for the first sentence; “Each candidate has one or more qualifications”. From Fig. 6 it is clear that the noun “Qualification” does not play the role of the sentence subject in the requirements. Consequently, the noun is not mapped into an entity.
Rule 7: When a noun meets the following conditions, it represents an entity.
1. The noun does not represent the object of a sentence.
2. The noun does not represent the subject of a sentence.
Example 4: A student registers on many modules. The school keeps the following information about each student: name, date of birth, nationality, address, and department. Fig. 7 shows Stanford Dependency of example 4.

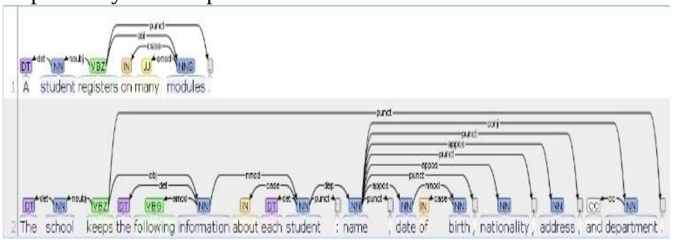

1 A student registers on many modules
2 The school keeps the following information about each student : name , date of birth , nationality , address , and department .

Fig 7: Stanford Dependency of Example 4

According to information obtained from Fig. 7, the common noun “Department” plays neither the role of sentence subject nor sentence object, but it is an entity because there are many departments in the school.
The training set was also trained using decision tree classifiers: TreeJ48, Random Tree, and RepTree. Fig. 8, Fig. 9 and Fig. 10 demonstrate the decision trees obtained at the end of their training.

```
Frequency = No: No                          1
Frequency = Yes                             2
|  Common Noun = No: No                     3
|  Common Noun = Yes                        4
|  |  Sentence Object = No: Yes             5
|  |  Sentence Object = Yes                 6
|  |  |  Sentence Subject = No: No          7
|  |  |  Sentence Subject = Yes: Yes        8
```

Fig 8: The J48 tree

```
Strong Entity = No                               1
|  Frequency = No                                2
|  |  Common Noun = No : No                      3
|  |  Common Noun = Yes                          4
|  |  |  Sentence Subject = No                   5
|  |  |  |  Sentence Object = No : No            6
|  |  |  |  Sentence Object = Yes : No           7
|  |  |  Sentence Subject = Yes : No             8
|  Frequency = Yes                               9
|  |  Sentence Object = No                       10
|  |  |  Common Noun = No : No                   11
|  |  |  Common Noun = Yes                       12
|  |  |  |  Sentence Subject = No: Yes           13
|  |  |  |  Sentence Subject = Yes : No          14
|  |  Sentence Object = Yes : No                 15
Strong Entity = Yes                              16
|  Frequency = No                                17
|  |  Sentence Subject = No : No                 18
|  |  Sentence Subject = Yes : No                19
|  Frequency = Yes                               20
|  |  Common Noun = No : No                      21
|  |  Common Noun = Yes                          22
|  |  |  Sentence Subject = No                   23
|  |  |  |  Sentence Object = No : Yes           24
|  |  |  |  Sentence Object = Yes : No           25
|  |  |  Sentence Subject = Yes                  26
|  |  |  |  Sentence Object = No : Yes           27
|  |  |  |  Sentence Object = Yes : Yes          28
```

Fig 9: The Random tree

```
Frequency = No : No                  1
Frequency = Yes                      2
|  Common Noun = No : No             3
|  Common Noun = Yes : Yes           4
```

Fig 10: The RepTree

Fig. 8 shows that the TreeJ48 classifier depended on frequency as a distinguishing feature between entity nouns and other nouns. Fig. 8 shows a decision tree that uses noun features to distinguish entity nouns from other nouns. According to Fig. 8, a noun is mapped into an entity if it satisfies the following conditions:

1. The noun has a frequency, is a common noun, and does not play the role of a sentence object in the requirement definition document.
2. The noun has a frequency, is a common noun, and it reflects a sentence object and subject role within the condition definition document.

Fig. 8 indicates that a noun is not an entity in the following situations:
1. There is no frequency for the noun (the noun occurs only once in the requirement definition document).
2. While the noun is frequent, it is not a common noun.
3. The noun is frequent, is a common noun, and represents a sentence subject but not a sentence object.

Example 5: In the sentences, "A university has many schools. Each school has many departments. A department includes many modules. Student must register in a department to study its modules", Fig. 11 shows POS taggers for Example 5, and Fig.12 illustrates Stanford Dependency for Example 5.

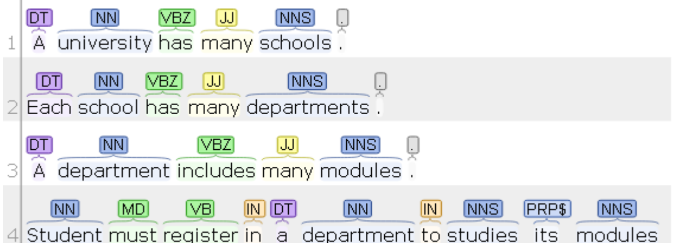

1 A university has many schools .
DT NN VBZ JJ NNS .
2 Each school has many departments .
DT NN VBZ JJ NNS .
3 A department includes many modules .
DT NN VBZ JJ NNS .
4 Student must register in a department to studies its modules
NN MD VB IN DT NN IN NNS PRP$ NNS

Fig 11: POS Taggers for example 5

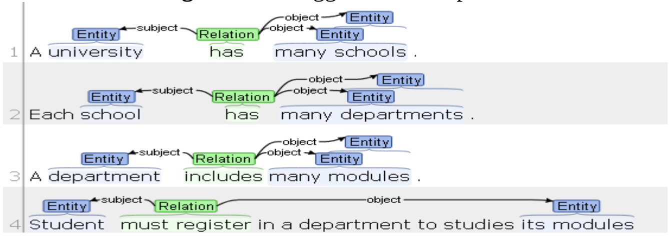

1 A university has many schools .
Entity, subject, Relation, object, Entity, object, Entity
2 Each school has many departments .
Entity, subject, Relation, object, Entity, object, Entity
3 A department includes many modules .
Entity, subject, Relation, object, Entity, object, Entity
4 Student must register in a department to studies its modules
Entity, subject, Relation, object, Entity

Fig 12: Stanford Dependency for example 5

A NN tag means the noun is a common noun. Thus, University, school, department modules and student are common nouns as presented in Example 5. A university is a common noun and with no frequency, as it is mentioned only once in Example 5. As a result, it is not an entity according to layer number 1 in Fig. 8 which shows if the noun is not repeated (no frequency), then the noun is not an entity. In Example 5, "school" was listed twice as a common noun. It has a frequency of two. According to the decision tree depicted in Fig. 8, and since there is a frequency of two, it will be transferred from layer one to layer two. From layer two "school" moved to layer number four, by-passing layer 3, because it is a common noun. From layer number four, "school" moved to layer number six, by-passing layer number five because it plays the role of sentence subject in sentence number one. From layer number six "school" is moved to layer number eight. Layer number seven is ignored because the noun "school" is a sentence subject in sentence number two, so it will be given a "yes" which means the noun "school" is mapped into an entity. In similar manner the J48 Tree deals with other nouns such as a department, module, and student.
Except in following situations the Random Tree classifier assigns a noun as non-entity:
1. Strong entity=No and Frequency=Yes and Sentence object=No and Common noun=Yes and Sentence subject=No ➔Entity=Yes.
2. Strong entity=Yes and Frequency=Yes and Common noun=Yes and Sentence subject=No and Sentence object=No ➔Entity=Yes.
3. Strong entity=Yes and Frequency=Yes and Common noun=Yes and Sentence subject=Yes and Sentence object=No ➔Entity=Yes.
4. Strong entity=Yes and Frequency=Yes and Common noun=Yes and Sentence subject=Yes and Sentence object=Yes➔Entity=Yes.

Returning to the common noun "school" presented in Example 5. By checking Fig. 11 and Fig. 12 following features are assigned to the noun: Strong entity=Yes, Frequency=Yes, Common noun=Yes, Sentence subject=Yes and Sentence object=Yes. The values attached to the features assign the noun as an entity.
Apart from following situation the RepTree classifier represented in Fig. 10 assigns a noun as non-entity:
Frequency =Yes and Common noun =Yes->Entity=Yes.
By checking Fig. 11 and Fig. 12, the following features are assigning to the noun:
Strong entity=Yes, Frequency=Yes, and Common noun=Yes then the classifier assigns the noun "school" as an entity.
The rules obtained from the Ripper, PART, and Decision Tree classifiers were evaluated using the test set. The aim is to determine how accurate the rules are in distinguishing entity nouns from other nouns. Linguistic rules for translating nouns into entities are as follows:

LR1= A common noun is mapped into an entity [7], [30].
LR2= A sentence subject represents an entity [9].
LR3= A sentence object is transferred to an entity [9].
LR4= A strong entity is mapped into an entity [11].
LR5= Frequency of a noun is a sign indicating that it represents an entity [12].

The test set was also used to test the linguistic rules, LR1 to LR5. The comparative outcomes of testing the test set using machine learning and linguistic rules, are presented in Table 3.

Table 3: A comparison of the outcomes obtained from testing the test set on machine learning rules and linguistic rules.

Rule NO	IC	CP	IP	Accuracy	MI
MLR1	11	10	1	90%	292
MLR2	133	112	21	84%	170
MLR3	177	132	45	75%	126
MLR4	125	111	14	89%	178
MLR5	59	59	00	100%	244
MLR6	06	05	01	83%	297
MLR7	143	143	00	100%	160
J48 Tree	303	257	46	85%	00
Random Tree	303	257	46	85%	00
REPTREE	303	257	46	85%	00
LR1	244	154	90	59%	59
LR2	10	10	00	100%	293
LR3	07	01	06	14%	296
LR4	179	132	47	73%	124
LR5	154	154	00	100%	149

Table keys:
IC: Instances count
CP: Correct Prediction
IP: Incorrect Prediction
MI: Missed Instances

The majority of machine learning rules are compound rules derived from the RIPPER, PART, and Decision tree classifiers. Table 2, Fig. 8, Fig. 9, and Fig. 10 demonstrate the machine learning rules. Linguistic rules are examples of rules found in the literature and are presented in Section Two. Linguistic rules are a "single set of rules", referring to the fact that only one feature maps a noun to an entity.
Compound rules, which include machine learning rules, require several features in order to map a noun to an entity. Table 3 contains six columns. The first column is the rule number. MLR1–MLR7 as represented in Table 2. Fig. 8, Fig. 9, and Fig. 10 show the J48 Tree, Random Tree, and RepTree, respectively. LR1 to LR5 are linguistic rules. The number of instances that meet the requirements of the rules found in the test set is expressed by the "Instances count column". The number of times the rules predicted correct values is shown in the "Correct prediction column". The number of times the rules predict incorrect values is shown in the "Incorrect prediction column".
The "Accuracy" column presents the rule's accuracy, which is the

correct prediction value divided by the number of instances. Missed instances is the number of instances when the rule is not able to make prediction. According to the results presented in Table 3, using decision tree classifiers greatly increases the ability to identify entities. The accuracy range is 85 percent and there are no missing instances. The accuracy is between 75-100 percent when the RIPPER classifier and the PART classifier are used. However, the number of missed instances increased dramatically. The accuracy was between 14 and 100 percent when the linguistic rules were applied, but the number of missed instances was also high.

Conclusion

In order to map natural language text that describes a problem statement of a specific domain into a conceptual model, it is necessary to analyze the text. Alternatively, there is a need for accurate rules for the transmission of natural language text into a conceptual model. Since the creation of the entity relationship model in 1976, a number of rules have been proposed for mapping the text of the problem statement.

However, the rules for mapping a noun to an entity are not comprehensive, and there is no rule that is always applicable. Common nouns, sentence subjects and objects can all be assigned to entities, but not every common noun, sentence subject, sentence object, strong entity noun or high-frequency noun can be assigned to an entity. To solve this problem, machine learning classifiers are used for the development of a novel set of rules for extracting entities from natural language text. Ten rules have been developed using the Ripper, PART and Decision Tree classifiers. Performance comparisons were made between linguistic rules collected from the literature and the rules of machine learning. The results show a dramatic improvement when machine learning was used. In addition, rules that are based on decision tree classifiers are better than those based on the RIPPER and PART classifiers for predicting entities from other nouns. These rules can support student and novice designers in extracting entities within the text of a problem domain. Development of a set of rules for extracting relationships between entities is a direction for future research.

References

[1]- M. Javed and Y. Lin, "iMER: Iterative process of entity relationship and business process model extraction from the requirements," *Inf. Softw. Technol.*, vol. 135, p. 106558, 2021.

[2]- I.-Y. Song, K. Yano, J. Trujillo, and S. Luján-Mora, "A taxonomic class modeling methodology for object-oriented analysis," in *Information Modeling Methods and Methodologies: Advanced Topics in Database Research*, IGI Global, 2005, pp. 216–240.

[3]- I.-Y. Song, Y. Zhu, H. Ceong, and O. Thonggoom, "Methodologies for semi-automated conceptual data modeling from requirements," in *International Conference on Conceptual Modeling*, 2015, pp. 18–31.

[4]- M. Luisa, F. Mariangela, and N. I. Pierluigi, "Market research for requirements analysis using linguistic tools," *Requir. Eng.*, vol. 9, no. 1, pp. 40–56, 2004.

[5]- C. J. Neill and P. A. Laplante, "Requirements engineering: the state of the practice," *IEEE Softw.*, vol. 20, no. 6, pp. 40–45, 2003.

[6]- M. Omar, "Semi-Automated Development of Conceptual Models from Natural Language Text," University of Huddersfield, 2018.

[7]- P. P.-S. Chen, "English sentence structure and entity-relationship diagrams," *Inf. Sci. (Ny).*, vol. 29, no. 2–3, pp. 127–149, 1983.

[8]- M. Omar and A. Abdulla, "The Entities Extraction for Entity Relationship Models from Natural Language Text via Machine Learning Algorithms," 2020.

[9]- V. B. R. V. Sagar and S. Abirami, "Conceptual modeling of natural language functional requirements," *J. Syst. Softw.*, vol. 88, pp. 25–41, 2014.

[10]- G. A. Miller, *WordNet: An electronic lexical database*. 1998.

[11]- O. Thonggoom, "Semi-Automatic Conceptual Data Modeling Using Entity and Relationship Instance Repositories," Drexel University, 2011.

[12]- M. A. Omar, "Performance Evaluation Of Supervised Machine Learning Classifiers For Mapping Natural Language Text To Entity Relationship Models," *J. Pure \& Appl. Sci.*, vol. 20, no. 1, pp. 6–10, 2021.

[13]- E. B. Hunt, J. Marin, and P. J. Stone, "Experiments in induction.," 1966.

[14]- L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[15]- S. K. Murthy, "Automatic construction of decision trees from data: A multi-disciplinary survey," *Data Min. Knowl. Discov.*, vol. 2, no. 4, pp. 345–389, 1998.

[16]- J. R. Quinlan, "C4. 5: Programs for Machine Learning." Morgan Kaufmann Publishers Inc., 1993.

[17]- J. R. Quinlan, "Discovering rules by induction from large collections of examples," *Expert Syst. micro Electron. age*, 1979.

[18]- T.-S. Lim, W.-Y. Loh, and Y.-S. Shih, "A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms," *Mach. Learn.*, vol. 40, no. 3, pp. 203–228, 2000.

[19]- S. Ruggieri, "Efficient C4. 5 [classification algorithm]," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 2, pp. 438–444, 2002.

[20]- J. Fürnkranz, "Separate-and-conquer rule learning," *Artif. Intell. Rev.*, vol. 13, no. 1, pp. 3–54, 1999.

[21]- S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: a review of classification and combining techniques," *Artif. Intell. Rev.*, vol. 26, no. 3, pp. 159–190, 2006.

[22]- S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," *Informatica*, vol. 31, pp. 249–268, 2007.

[23]- J. Kivinen, "Online learning of linear classifiers," *Adv. Lect. Mach. Learn.*, pp. 235–257, 2003.

[24]- D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," 1985.

[25]- D. Saad, *On-line learning in neural networks*, no. 17. Cambridge University Press, 2009.

[26]- J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu, "Learning Bayesian networks from data: An information-theory based approach," *Artif. Intell.*, vol. 137, no. 1–2, pp. 43–90, 2002.

[27]- Y. Yang and G. I. Webb, "On why discretization works for naive-bayes classifiers," in *Australasian Joint Conference on Artificial Intelligence*, 2003, pp. 440–452.

[28]- R. R. Bouckaert, "Naive bayes classifiers that perform well with continuous variables," in *Australasian joint conference on artificial intelligence*, 2004, pp. 1089–1094.

[29]- Y. Qin *et al.*, "Entity Relation Extraction Based on Entity Indicators," *Symmetry (Basel).*, vol. 13, no. 4, p. 539, 2021.

[30]- A. M. Tjoa and L. Berger, "Transformation of requirement specifications expressed in natural language into an EER model," in *International Conference on Conceptual Modeling*, 1993, pp. 206–217.