Semantic Based Database Schema Matching

Hiba Jlilati

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
June 2020
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Ali Hakan Ulusoy
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Işık Aybay
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Duygu Çelik Ertuğrul
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Duygu Çelik Ertuğrul
2. Asst. Prof. Dr. Yiltan Bitirm
3. Asst. Prof. Dr. Mehtap Köse Ulukök

ABSTRACT

Schema matching is a process that takes multiple schemas as an input, applying a mapping between these schemas and produces one schema with the matching components. Most of data sources of E-business companies are heterogeneous, which making it difficult task for the integration and the exchange of the data. The purpose of this thesis is to indicate a matching system that involves required steps to produce a map while matching relational databases to ontology. According to researchers schema matching using two main approaches for classification; Individual Matcher Approaches and Combining Matcher Approaches with multiple levels of matching criteria. In this thesis individual matcher based on schema with linguistic based is used. A research done to find the suitable methodology that works effectively with schema matching; Conventional Neural Network (CNN) indicated to extract the important features and classify it to get a match. CNN uses word-embedding representations of two words as inputs to derive the semantic characteristics between the two words and provide a score as the result of how likely they fit the CNN pattern, Cosine Similarity and Jaro Winkler algorithms implemented with CNN to get better results. The combination of semantic schema matching with machine learning algorithm provide great improvement in the matching field, which become easier and time consuming. The performance of the proposed system showed that semantic similarity scores with clustering using CNN model could produce more than 90% accuracy. NetBeans and protégé are used to build the proposed system.

Keywords: Semantic Matching, Ontology, Machine Learning, CNN, Word Embedding, Backpropagation.

ÖZ

Şema eşlemesi, birden çok şemayı girdi olarak alan, bu şemalar arasında bir eşleme uygulayan ve eşleşen bileşenlerle yeni bir şema üreten işlemdir. E-ticaret şirketlerinin veri kaynaklarının çoğu heterojendir ve bu da verilerin entegrasyonu ve değişimi için zor bir yapıdır. Bu tezin amacı, ilişkisel veri tabanlarını ontolojideki kavramlar ile eşleştirirken, bir harita üretmek için gerekli adımları içeren bir eşleştirme sistemini belirtmektir. Sınıflandırma için iki ana yaklaşım kullanan araştırmacı şema eşleşmesine göre; Bireysel Eşleştirici Yaklaşımları ve Eşleştirici Yaklaşımlarını çok sayıda eşleştirme kriteri ile birleştirmektir. Bu tezde dil bilimsel şemaya dayalı bireysel eşleştirici yöntemi kullanılmıştır. Şema eşleşmesi ile etkili bir şekilde çalışan uygun metodolojiyi bulmak için yapılan bir araştırmada, Konvansiyonel Sinir Ağının (CNN) önemli kavramsal özellikleri çıkarması ve yeni bir eşleşme elde etmek için sınıflandırması uygulanmıştır. CNN, iki sözcüğün anlamsal bağlılıklarını bulmak için, CNN modeline, Kosinüs Benzerliğine ve Jaro Winkler algoritmalarına eşleşme olasılıklarının üzerinden iki sözcüğün eşleşmesini tanımlar. Anlamsal şema eşleşmesinin makine öğrenme yaklaşımı ile birleşimi, veri eşleştirme sahasında daha kolay ve zamandan kazanç sağlayacak büyük gelişme sağlar. Önerilen sistemin performansı, CNN modeli ve anlamsal benzerlik algortimaları uygulanarak, sistem performansının doğruluk oranını %90'dan fazla üretebileceğini göstermiştir. NetBeans ve Protégé araçları önerilen sistemi oluşturmak için kullanılmıştır.

Anahtar Kelimeler: Semantik Eşleme, Ontoloji, Makine Öğrenmesi, CNN, Kelime Gömme, Geri çoğaltma.

DEDICATION

For my parents and friends.

ACKNOWLEDGMENT

I would like to thank my family for their love and support that they gave to me without them I couldn’t accomplish any of this, and a special thanks to my thesis supervisor Assoc. Prof. Dr. Duygu Çelik Ertuğrul from Eastern Mediterranean University's, which helped me and was there for me whenever I needed help, thank you for your patients and your endless support and guidance to show me the right direction to write this thesis and make it correctly done.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURE

LIST OF SYMBOLS AND ABBREVIATIONS

ANS	Artificial Neural Network System
ASCII	American Standard Code for Information Interchange
BP	Back Propagation
CNN	Convolutional Neural Network
CO	Compound Words
COMA	Combination of Matching Algorithms
GLUE	Generalized Likelihood Uncertainty Estimation
NLP	Natural Language Processing
NOM	Naive Ontology Mapping
NSS	Node Semantic Similarity
OWL	Web Ontology Language
PCA	Principal Component Analysis
POS	Part Of Speech
QOM	Quick Ontology Mapping
RDF	Resource Description Framework
SEMINT	SEMantic INTegrator
SF	Similarity Flooding
SOM	Self-Organizing Map
STS	Semantic Text Similarity
SW	Semantic Web
VSM	Vector Space Model
WE	Word Embedding
WN	Word Net

Chapter 1

INTRODUCTION

Schema matching is a process that takes multiple schemas as an input, applying a mapping between these schemas and produces one schema with the matching components. Many researchers consider the problem of matching schemas as an obstacle for semantical fusion [1]. Where a huge number of matching algorithms have developed since. According to many researchers who have specified that matching schemas is the way of determining meanings between components of metadata such as schemas of the databases, ER diagrams and sematic-based database. It is necessary and critical for the interoperability and integration of data in different applications, such as data storage, Web sources integration and alignment of ontology in the semantic web. Actually, the process of layout matching has been improved from manual to semi-automatic [2-3].

Furthermore, schema-matching using two main approaches for classification; Individual Matcher Approaches and Combining Matcher Approaches. Individual matcher based on two matching criteria, Schema-Only which have two levels: Element Level where the matching could be either linguistic based such as name similarity or constraint based such as type similarity and Structural Level where the matching is constraint based like graph matching. The other matching criteria is Instance based which has one element level, where the matching could be either linguistic or constraint based. While the combining matcher approach based on two matching

criteria as well Hybrid Matcher and Composition Matcher that could be manual or automatic [4]. In this study individual matcher based on schema with linguistic based is used.

The purpose of this thesis is to indicate a matching system that involves required steps to produce a map while matching relational databases to ontology. A research is done to find the suitable methodology that works effectively with schema matching; Convolutional Neural Network algorithm is used because of its effectiveness to map random and heterogeneous databases, with the use of Cosine Similarity and Jaro Winkler algorithms.

The database that used in this thesis is a data of four real e-commerce companies N11, Mikro, Logo and Serotonin. Which contain multiple tables but one table is used from each firm. Each table have a different number of fields. In this study, 302 fields have been used, for each field there is an expected name that used in testing. The performance of the system after testing showed that semantic similarity scores with clustering using CNN model could produce more than 90% accuracy.

In this thesis, the background about the algorithms that used is explained in Chapter 2. Chapter 3 presents the literature review and other researcher's related works. The database that used in this study explained in Chapter 4. Chapter 5 covers the implementation of the system. In chapter 6 the used methodologies are present and explained in details. Chapter 7 presents the system interfaces. In Chapter 8 the testing results are displayed with a compression with other systems. The conclusion is in Chapter 9.

Chapter 2

BACKGROUND

2.1 Background about Schema Matching

A schema is a collection of components connected by a few structures, such as XML construction, SQL pattern, substance relationship graphs, ontology depictions and definitions of the interfacing. Schema matching is a process that takes two heterogeneous schemas as an input and generates a set of mappings as an output.

According to Rahm and Bernstein [4] types of pattern coordinating approaches was displayed, which distinguished between individual and combinational matchers, schema and instance matchers, and matching approaches based on language and constrain. In the following part, these approaches discussed.

2.1.1 Individual vs. Combinational

Individual matchers: The matching method done with a single algorithm by an individual matcher.

Combinational matchers: Two sorts of combinational coordinating can be used for combinational matchers, which are hybrid matchers take into account several requirements for performing the matching function and composite, where matching algorithms run separately on two schemes and merge the outcome.

2.1.2 Schema vs. Instance

Schema-based matchers: In this kind of matcher metadata, data types, component names and basic properties and/or models have considered.

Instance/content-based matchers: Content-based matcher takes data and data content into consideration; this combines the methods of artificial intelligence and data mining techniques. A clustering-based pattern coordinating approach that make the recall better and exactness rate of strategy coordinating by computing more exact scores [5].

A proficient content-based approach for optimizing the coming about of pattern coordinating. It is possible to work with other pattern coordinating methods of freely [6].

2.1.3 Linguistic vs. Constraint

Linguistic matchers: The linguistic matching method takes into account the name and textual explanation of the labels or components in the schema. Different methods including Edit Distance [7], N-gram and Sound-EX [8] (indexing names by sound, as pronounced in English) are used in the linguistic approach [9].

Constraint-based matchers: The limitation of the methodology regarded to the component limitations are the uniqueness of the data types and keys.

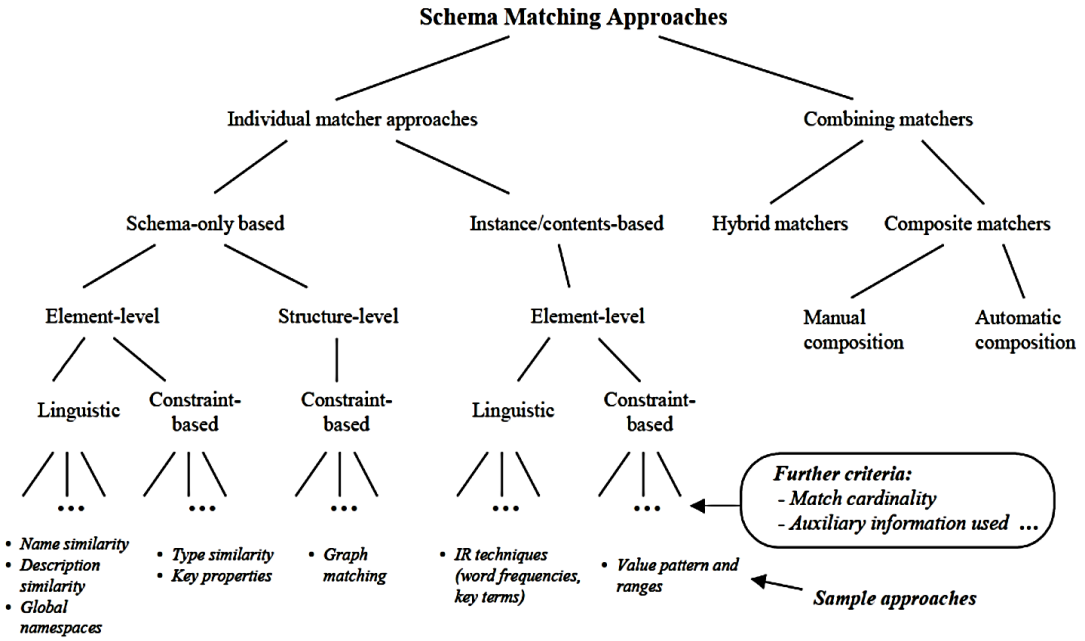

Figure 1: Schema Matching Approaches

2.2 Web Ontology Language

According to McGuinness and Van Harmelen [10], Web Ontology Language (OWL) used when applications need to process the information in the documents, to presented to people, it can be used expressly to describe the significance of words in vocabulary and their interactions. These concepts called ontology and their interrelationships. Furthermore, OWL has more means of communicating significance and semantics than XML (Extensible Mark-up Language), RDF (Resource Description Framework) and RDF-Schema. So, OWL extends beyond these forms in its capacity to display machine-readable material on the Web.

Also described that ontologies include computer-usable descriptions of fundamental ideas in the domain and their interactions. The authors concluded that the Web requires ontologies with a substantial degree of framework. These details need to be specified for the following ideas: classes and relationships that may occur between items, these items may have characteristics.

They also indicated in their research that ontologies notably represented in the evolving semantic-web as a manner of characterizing document semantics and allowing web applications and smart agents to use semantics. It also can be very helpful to structure and define the significance metadata concepts that are being gathered and defined for a society. Using ontologies' apps may be "smart," if they can function more precisely on the conceptual stage of humans. Ontologies are critical for apps that searched or merged data from various groups.

Agreeing to McGuinness and Van Harmelen [10], web ontology language contain a lot of feature that helps us to deal with it which related to RDF schema. Such as:

- Class: a class describes a group of individuals that belong to each other because they shared the same properties.
- Sub Class Of: can organize class in a specialization hierarchy way.
- Property: Properties can used to state relationships between entities or from individuals to data values, Object Property and Data Type Property are two of them.
- Individual: Individuals are instances of classes, and properties may use to connect individuals to each other. OWL also requires property limitations to demonstrate how properties can used by instance of a class there are two forms "all Value Form, some Value Form".

2.3 Convolutional Neural Network

This machine-learning algorithm becomes the researchers main focused algorithms in many ranges like visual identification, image interpretation and retrieval of natural language. CNN are an extension of the ordinary Neural Networks inspired by the vision processing in living organisms [11]. Designed to work with two-dimensional

image data, while one-dimensional and three-dimensional data can be used as well [12]. The benefits of CNNs are the spatial invariance and automated creation of features resulting from the use of trained filters for the phase of convolution and pooling. [11], CNN used everywhere and it does an amazing job of image retrieval, opinion polarity detection, sentiment analysis and this kind of algorithms used in self-driving cars by doing parallel processing and able to train and test these Convolutional Networks efficiently.

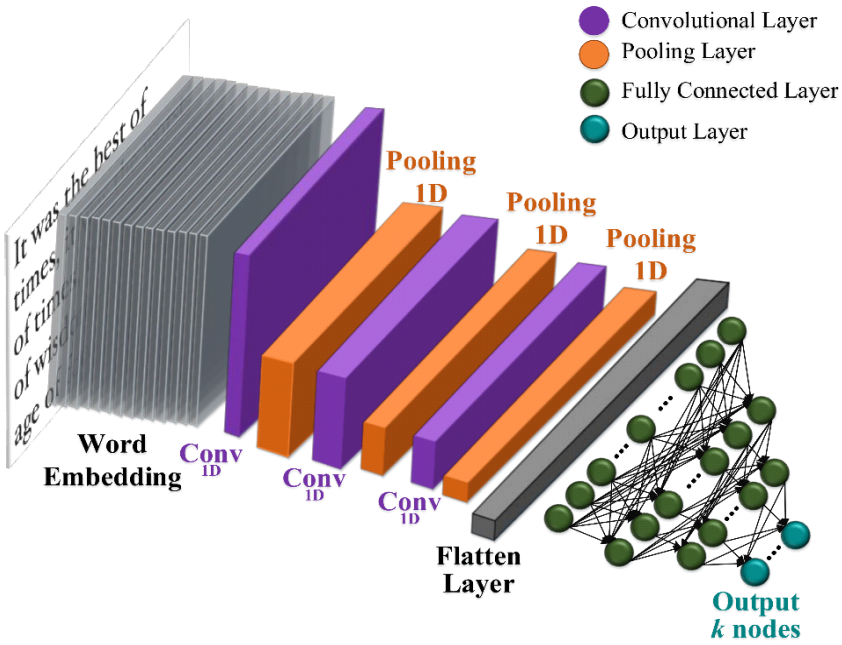

Figure 2: Architecture of CNN Layers

Figure 2 shows the architecture of CNN where it's basically had three layers:

- Convolutional layer: The convolution layer generates feature maps by applying a dot product between the input values and a filter, where the filter is a matrix with predefined dimensions that generates randomly.
- Pooling layer: It is a combination of multiple vectors and result of one dimension vector by choosing the max or the average values that exist in different dimensions, the most common strategy is max-pooling where the

maximum number of the window is chosen, the convolution and pooling layers repeating many times, and result a set of feature maps which then it condense into a one vector.

- Fully connected layer: it collects all the results from the previous layers and resulting the output.

2.4 Cosine Similarity

The cosine similarity is a function that calculate the similarity between two strings by finding the angle between two vectors A, B and resulting a non-negative value in the range [0, 1]. When the resulted value is 1 that's mean the two words are exactly match, if it's between the range [0, 1] that's mean the strings either synonym when the value close to 1 otherwise it's not matched.

$$\text{Cos}\,\theta \frac{A \cdot B}{\| A \| \| B \|} \tag{1}$$

2.5 Jaro Winkler

A function that finds the exact weight between two strings to get the correct spelling of the input word comparing with a dictionary of words. Which done character by character taking in to account the length of the two strings S1, S2 and the number of transpositions T equals to the number of the mismatch characters between the two strings divided by 2, which done by formula (2).

$$d_j = \begin{cases} 0 & if\ m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & otherwise \end{cases} \tag{2}$$

Chapter 3

LITRATURE REVIEW

3.1 Introduction

In this chapter related work about schema matching and convolution neural network that done by other researchers have been evaluated to find the suitable method to use it for matching schema with machine learning techniques. Previous work indicates that numerous algorithms and methods proposed to match the schemes ' attributes and elements.

3.2 Related Work of Schema Matching

Islam and Inkpen provided an algorithm called it Semantic Text similarities for calculating paragraphs semantic similarity by using the measures of a corpus-based of the similarities of a semantic word with Longest Popular Subsequence (LCS). The authors concentrate on measuring comparisons between two phrases or two short texts, by using three functions to calculate the similarities, first string similarity it is done by using three adjusted versions of LCS and taking a weighted total of them; second semantic word similarities it is done using corpus-based calculations. Finally, an appropriate common-word similarity feature used to insert syntactic knowledge in their system. By the combination of the three functions with *normalized longest common subsequence* (NLCS) **(3)**, the semantic similarities found.

$$\mathrm{NLCS}(r_i, s_j) = \frac{length\left(LCS(r_i, s_j)\right)^2}{length(r_i) \times length(s_j)} \tag{3}$$

The result of the algorithm was a score in a range between 1 and 0, the result of the algorithm was good by using 4,076 training and 1,725 test pairs of Microsoft paraphrase corpus [1].

COMA system providing a structure to contain both obtained results and evaluation of the effectiveness of the matchers [9], Developed a system by combining multiple algorithms for matching XML and relational schemas, it takes two schemas as an input and results a group of mapping elements with a similarity value between [0, 1]. When its 0 it's meant the two elements are strongly dissimilar, when its 1 it's meant the two elements are strongly similar. Also, it produces element-level matches of 1:1 local and m:n global cardinality.

NOM adopts COMA's concept of integrated alignment where it based on the rules in the collection of elementary matchers, using specifically codified information in ontologies, like details about super and sub properties or concepts. Its system meets 17 Regulations. For example, one of the regulations notes that, where super concepts are the same, the real concepts are identical to one another. NOM also exploits a collection of techniques focused on instances [13].

Simi-Match: The process exploits the overlap that happened between data by mapping between the elements of the sources and global schema. This involves extracting the semantically distinct properties from the sources, regardless of their model or namespace, and transfer the output into the virtual view. This process repeated, incrementally building up a virtual view, ignoring duplicate information, to create a unique set of semantically distinct properties from all of the sources [14].

Li-Jun Chen Performed deep web matching system uses concept-word of attributes and its mechanical matching are naïve but very efficient and effective aggregation of synonymous attributes. In addition, semantic heterogeneous model, that is consist of three matchers; semantic matcher where its match the attribute according to its meanings using WordNet, character matcher where it's done by matching the characters inside the name of the attributes using Edit-distance, instance matcher and their combinatorial method the system architecture show in in figure [15].

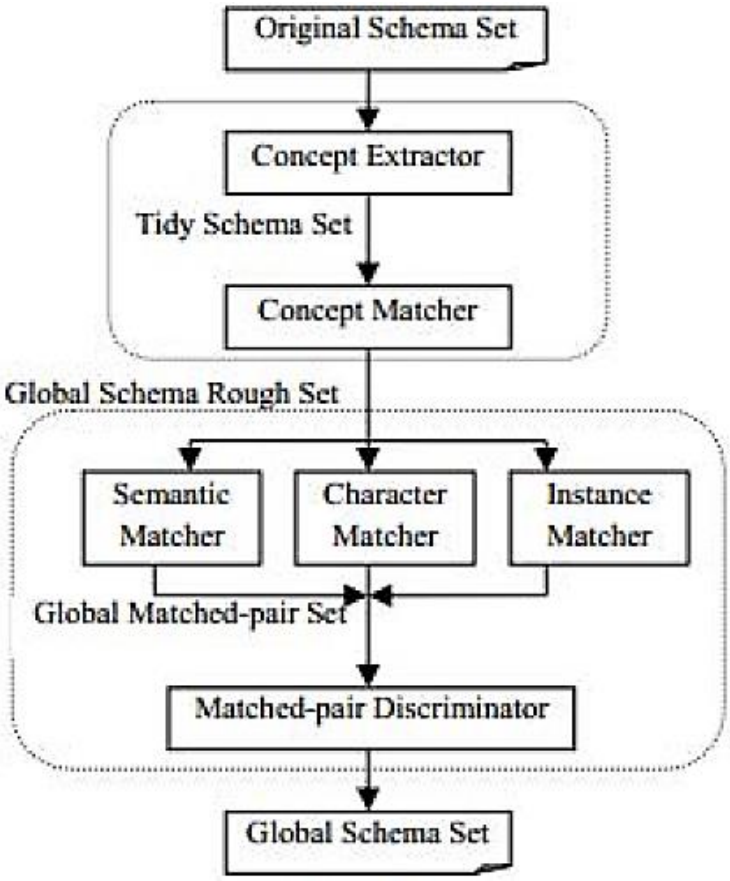

Figure 3: Deep Web System Architecture

Ningsheng Jian [16] Developed a tool for aligning ontologies named it FalconAO, which had two matchers, LMO matching based on Linguistic and GMO matching based on graph matching ontologies. Where represented the ontologies by direct graphs in a way to find the similarities structure of these graphs to achieve linguistic

similarity. LMO incorporated two different approaches: lexical comparison and statistical analysis where uses VSM in implementation.

Generic Schema Matching with Cupid developed by Madhavan [17], which discovered constraints in mapping between schema elements based on their data types and names. A few of the advance's technologies utilized were the precise utilize of etymological and basic coordinating, context-based matching of common sorts and an inclination towards the structure of the leaf in which the schema dwells. It generates 1:1 match or 1:n matchings.

Melnik, S provided a generic tool in a way to handle the deferent structures of data, which could be instances, schemas or a mix of the both. Their method done by converting the corresponding templates to direct labelled graphs, these graphs used in an iterative fix-point calculation. The results indicate which nodes in the first graph matched to other nodes in the second graph, when all the nodes in both graphs are similar so, the two graphs indicate as similar, because of this flooding of similarity this method called Similarity Flooding (SF); this tool needs the help of user to test and change the findings if necessary. Using GUI, the user changes the proposed match outcome by removing or inserting lines linking two schema components. The right match also relies only on the information that is accessible or understood by humans [18].

Sorrentino et al, [19] presented a NORMS method for normalization schematic labelling. Where it allows the extension of abbreviations and automatic enrichment of WN with CNs; it provides a GUI that assists the user in the standardization procedure and helps him to boost the results by removing the errors; it provides the possibility of

automatically correcting potential errors as an added feature; also ensures that the schema elements can dynamically annotated. Several matching programs can manipulate the output, NORMS takes many relational data as input like SQL, OWL, XML and MySQL; Using NORMS has provided a substantial increase in the number of right semantic relationships (synonym and hypernym) relationships found within heterogeneous relations, the entire cycle of normalization involves a small degree of interaction of human.

Researchers in the University of Washington [20] described GLUE system that uses machine-learning methods to map the semantics between ontologies' elements, the aim of the system was to find the matching terms between different ontologies using Joint probability distribution. The system finds the JPD for each term and uses the result to find suitable matching measure; to calculate the exact matching measures Jaccard coefficient (4) was used; if the two terms A,B are disjoint it takes the minimum value 0 and takes 1 as a maximum value when the two terms A, B are same.

$$\text{jaccard-sim}(A,B) = \frac{P(A \cap B) \,/\, P(A \cup B)}{\dfrac{P(A,B)}{P(A,B) + (A,\bar{B}) + (\bar{A},B)}} \tag{4}$$

Researchers [21] provided SEMINT a neural network-based method to help classifying the relationship of attributes in heterogeneous datasets reflecting the same definition of the real world would possibly have similar match in schema formats, restrictions, and patterns of data value; these similarities help us to recognize correspondences. The method first extracted the metadata (information of schema) from database using DBMS, the metadata will be normalized and will be the input for SOM algorithm as a classifier to classify attributes, the output of the classifier used to train neural networks to classify categories, then the related attribute will be classified

between databases. As shown in figure 2 the authors integrating semantic between two databases by extracting the metadata from these databases Faculty and Student by forming patterns describing their attributes, based on the patterns of attributes in Faculty and Student datasets, the trained neural network may classify corresponding attributes and find their similarities.

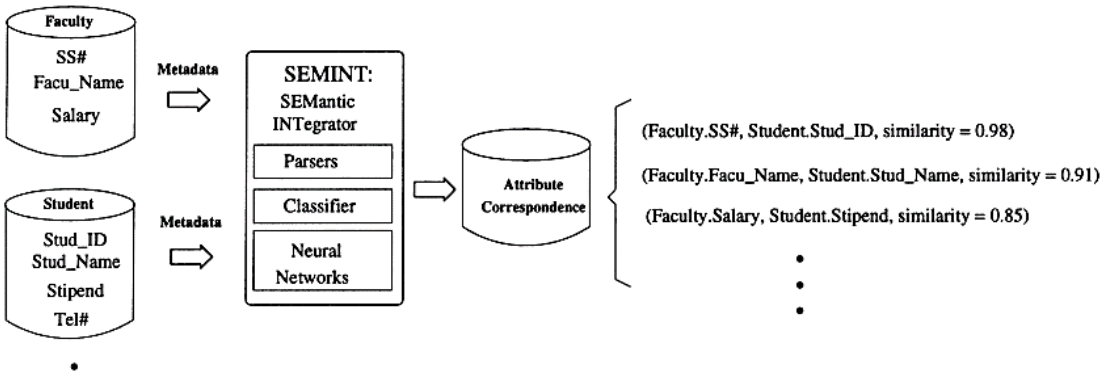

Figure 4: Overview of Semantic Integration In SEMINT

The following table depicts a few strategies and tools that classify this issue by utilizing lexical knowledge in various ways Table 1.

Table 1: Schema Matching Methods

year	Method discussed	Approaches	Storing schema	Schema used to match
Islam., & Inkpen, D [1]	Semantic Text Similarity (STS) method determines the similarity of two texts by combining string similarity, semantic similarity and common-word order similarity with normalization.	Schema-based	Common word order	Microsoft paraphrase corpus
Do and Rahm [9]	COMA system for Schema Matching approaches, providing a structure to contain both obtained results and evaluation of the effectiveness of the matchers.	Schema based	DBMS	Relational tables and rows or characteristics and components of XML
Shvaiko, Euzenat [13]	NOM (Naive Ontology Mapping) Based on rules, exploiting explicitly codified knowledge in ontologies, such as information about super- and sub-concepts, super- and sub-properties.	Instance based	Based on ontologies matching	Real world ontologies
Kettouch et al [14]	Simi-Match: The process exploits data by mapping between the elements of the sources and global schema. This involves extracting the semantically distinct properties from the sources, and transfer the output into the virtual view.	Semi-structured and LD (linked data)		UMBC tool constructed by combining Latent Semantic Analysis (LSA) word similarity and WorldNet.
Chen [15]	Performed on deep web matching uses concept-word of attributes, very efficient and effective aggregation of synonymous attributes. And semantic model which is consist of three matchers semantic matcher, character matcher, instance matcher, and their combinatorial method	Concept-word and semantic heterogeneous model		Finding the synonyms and selecting similarity values of attribute features.
Jian et al [16]	Developed FalconAO, a tool for aligning ontologies. LMO matching based on Linguistic and GMO matching based on graph matching ontologies integrated in FalconAO. It also uses VSM in implementation.	Schema based	Ontology based	Constructed Virtual document of an entity consists of "bag of terms" for each ontology entity

Madhavan, et al. [17]	Cupid method making context matches by exploiting views, keys and referential constraints. An automated element-based and structure-based matching called Linguistic matching. It generates 1:1 or 1:n mappings	Schema based	Database	XML schema with no shared elements
Melnik et al. [18]	Similarity Flooding (SF) Determining correspondence between nodes of the graphs using fix-point computation. Labeled graphs are being converted from schemas such as SQL and DDL relies on string similarity between element names.	Combined approach	18 schemas (XML and SQL)	Relational XML
Doan et al [20]	GLUE: Using machine learning techniques and automatic combination of match results to combine different matchers by compositing powerful approaches. Finding the exact match using Jaccard similarity to compute joint distribution.	Instance based	Ontology based	Database schemas
Li, W. S., & Clifton, C [21]	SemInt: Determining matching candidates using neural networks. Between two schemas individual attributes, a mapping gets created exploiting matching criteria up to 5 content-based and 15 constraint based.	Instance based	Database	Relational files

3.3 Related Work of CNN

An in-depth neural network that extracts information from character to sentence level (charSCNN) to evaluate short text feelings, utilizing two convolution layers to remove specific features from words and phrases of any duration. The network will easily explore the abundance of word embedding provided by unsupervised pre-training. They converted each term w into its word-level embedding by using the matrix-vector product in the embedding characters; generated local features around each word character and then combined them to create a fixed character-level embedding of the word using a max process. Also educated the network by minimizing the probability

of negative testing, utilizing stochastic gradient descent (SGD) to reduce the likelihood of negative logging [22].

Developed a novel approach for modelling short texts utilizing word embedding clustering and convolution neural networks, abbreviating them to (CCNN) clustering and CNN, clustering using density peaks to find the semantics in the embedding, Semantic units are used to measure Euclidean distance to locate their nearest word embedding (NWEs) with each semantic clique. Their CNN architecture has one convolutional and pooling layer, which has been taught under the cross-entropy target, configured iteratively by backpropagation (BP) with mini batches of samples [23].

Proposed for text classification tasks a Multiple Block Convolution Highways (MBCH) architecture. Some of these methods were implemented, such as highway network, Dense-Net, batch normalization and bottleneck layers, it achieved good results on benchmark's datasets and provided a new model Improved Word Vectors (IWV), which enhances pre-trained word embedding accuracy [24].

It is merging between CNN and bi-directional gated recurrent unit (BiGRU) with multi-attention mechanism, The CNN incorporated the basic classification system of the target sentiment to derive the subjective polarity in the sentence of the target keyword. BiGRU evaluated the sentiment polarity at the sentence level and the two factors combined to create a function vector for fusion, they used PCA (principal component analysis) dimensional reduction technologies to effectively reduce the global vector fusion function, obtaining a classification result combining two keywords and phrases [25].

Introduced Halo a user-friendly navigation program for hospital environments without the help of localization technologies, suggested a basic CNN for the identification of the point of departure and destination in the interface queries submitted, rather than a single soft-max output sheet, the origin and destination search term estimation was calculated utilizing two separate soft-max layers. The custom CNN model trained within a hospital using department names and key locations [26].

Majumder et al. developed a system for extracting personality traits from stream of consciousness essays focused on extractor CNN features, to obtain the sentence model in the form of n-gram function vectors; they feed sentences from the essays to convolution filters. Then each essay was interpreted by aggregating the vectors of its phrases, also they merged the vectors obtained with the Mairesse characteristics that were derived directly from the texts at the pre-processing stage to improve the performance of the system [27].

Chapter 4

DATABASE USED

4.1 Introduction

That data that used to test the system in this thesis are four datasets of real e-commerce firms; N11, Serotonin, Logo, Mikro. Later on, in this chapter, these four firms will be explained and discussed separately in details.

4.2 N11

The database N11 [28] is a real e-commerce database that contains many tables; Category, Cities, Shipment, Product, Products Service etc. In this thesis product table is used, the product table contains the following columns; field name, field description 1, field description 2, data type, length, mandatory and key. This table contains the firm's products information like product code, product order, product condition, product quantity etc.

Table 2: N11 Products Table

Field name	Field description 1	Field description 2	Data type	Length	mandatory	key
Product code			varchar	25	1	FK
Product order	Product image display order		varchar		1	
Product condition	Product State		varchar		1	
Product quantity	The amount of product		Varchar		1	

4.3 Serotonin

Serotonin is a real e-commerce database that contains many tables [29], in this thesis product table is used. The product table contains the following columns field name, data type, length, mandatory and key, this table contains the firm's products information like product code, stock piece, statement, group id, expiration date etc.

Table 3: Serotonin Products Table

Field name	Data type	Length	Mandatory	Key
Product_code	varchar	255	0	
Stock_piece	int	11	0	
Statement	text		0	
GroupId	varchar	20	0	FK
Expiration_date	varchar	50	0	

4.4 Logo

Logo is a real e-commerce database that contains many tables [30]; in this thesis, item table is used. The item table contains the following columns; field name, field description 1, field description 2, data type, length, mandatory and key. This table contains the firm's items information like item code, active, item photo, item producer code, tool etc.

Table 4: Logo Items Table

Field name	Field description 1	Data type	Length	Mandatory	key
ITEM-CODE	Material Code	varchar	25	1	PK
ACTIVE	Registration Status	int	2	1	
ITEM-Photo	picture	byte	1	0	
ITEM-PRODUCERCODE	Manufacturer Code	varchar	25	0	
TOOL	tool	byte	1	0	

4.5 Mikro

Mikro is a real e-commerce database that contains many tables [31]; Stocks table, Staff table, Sectors table, Producers table, Brands table, Packaging table etc. In this thesis, stocks table is used. The stocks table contains the following columns; field name, field description 1, field description 2, data type, length, mandatory and key. This table contains the firm's stocks information like stock code, max stock, stock order, stock type, stock orders day7 etc.

Table 5: Mikro Stocks Table

Field name	Field description 1	Field description 2	Data type	Length	mandatory	key
Stock-code	Product Code		varchar	25	1	PK
Max-Stock	Quantity of stocks		float		1	
Stock-order	Order time (days)		int			
Stock-type	Product Type		Tinyint		1	
Stock-orders day7	Order Days	Sunday	Smallint			

Chapter 5

SYSTEM PORPOSED

5.1 Introduction

The proposed system goes throw four stages to do the required matching, pre-processing stage, matching stage, system controller stage, CNN clustering stage. These stages will be explained and discussed in details. First in Section 5.2, the work cycle of the proposed system will be explained. In Section 5.3, the pre-process stage will be discussed. Then in Section 5.4, matching stage and how to save the data inside ontology will be explained. Finally, in Section 5.5 CNN clustering will explained.

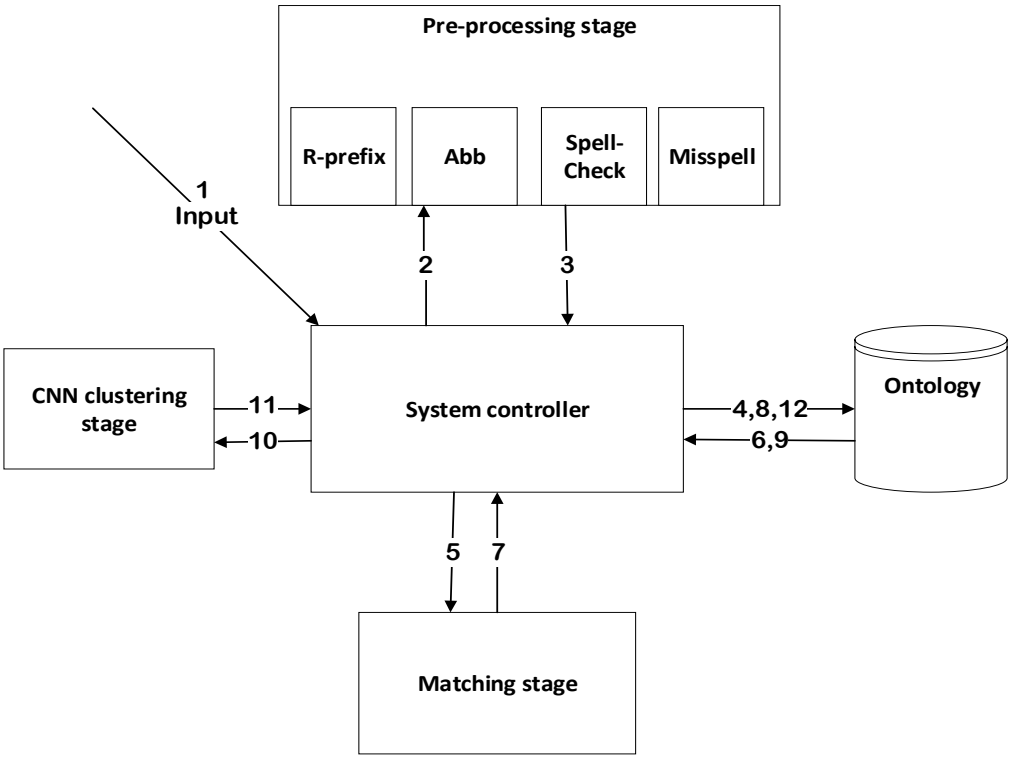

Figure 5: General Architecture of the System

5.2 System Controller

System controller responsible about the cycle of the data between the stages and controls the inputs and the outputs, it goes throw the following steps:

- Step1, when system runs the user enters data.
- Step2, system controller calls the pre-processing stage to process the data to be understandable for the system; it removes the prefix and split the words. Then it checks if the entered data have abbreviation words, it replaces it by using abbreviation dictionary. Alternatively, if the input data have misspelling errors its checks the spelling of the words using spelling dictionary and returns the correct form of the word. In addition, its match the synonyms.
- Step3, it returns the results to the system controller. Next, it checks if the ontology is empty then Step4 it saves the entered data as new record.
- Step (5-8), if the ontology is not empty, the system calls the matching stage to do the matching process between the input data and the data that been saved inside ontology and returns the matched records throw system controller to ontology to be saved.
- After that Step9, happened where the system controller extracts the saved data from ontology to be classified in CNN clustering stage Step10.
- Step11, the resulted data from CNN clustering stage will go to ontology using system controller to be saved Step12.

5.3 Pre-processing Stage

First, data should pre-processed before using, to be understandable for the proposed system, in the pre-processing stage underscores and prefix is removed by using the 'split' function, then handling the abbreviation words, misspelled words with checking the spell is explained.

5.3.1 Abbreviation

The following diagram displays the flow of the entered data when having abbreviation word, its open the abbreviation dictionary then check it line by line until finding a matching word and return the word with its full form.

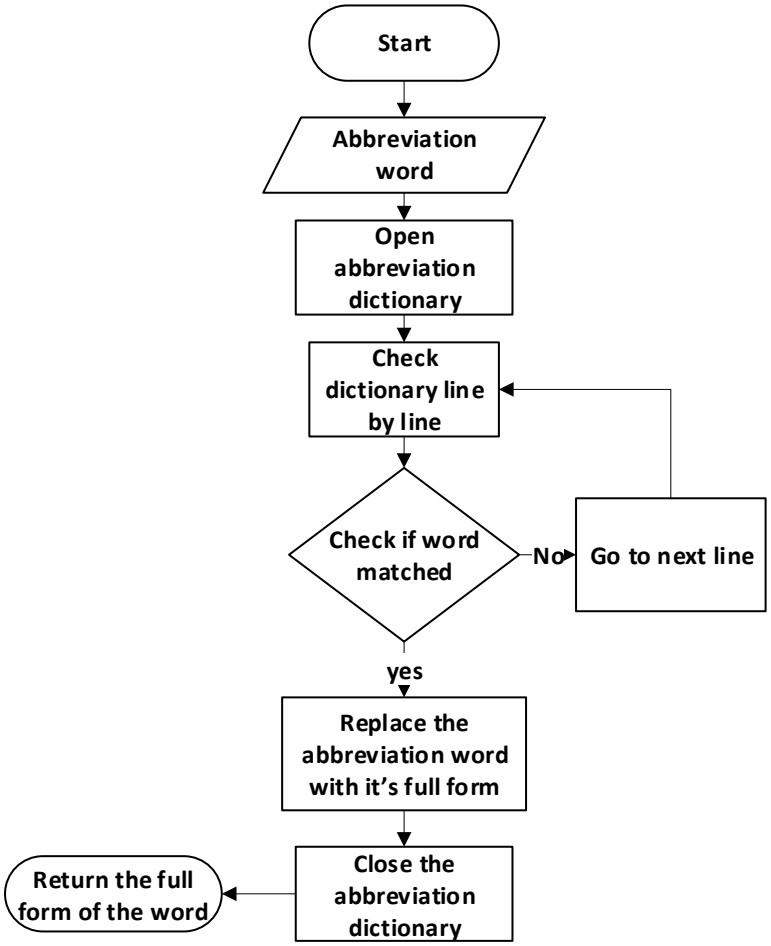

Figure 6: Flowchart of Abbreviation Process

5.3.2 Misspelling

The misspelling problem solved by applying Jaro Winkler. Table 6 shows the pseudo code of the misspelling process, where s_1, s_2 are the indexes of the first and second words, **m** is the number of the matched letters between the two words and **t** is the transposition value that increased when a mismatch occurred. Later on, the flowchart of the word inside the Jaro Winkler algorithm will displayed as well.

Table 6: Pseudo Code for Misspelling

1. function (s_1, s_2, t, m) 2. if s_1=0 or s_2=0 3. "string must not be null" 4. end if 5. mtp= match (s_1, s_2) 6. return j=((m/s_1+m/s_2+m-t/m)3) 7. end function 8. function match 9. for mi=0, mi<ms_1 10. for mii=0, mii<ms_2 11. if ms_1(mi)=ms_2(mii) 12. match ++, mi++, mii++	13. else mi++, mii++ 14. end if 15. end for 16. end for 17. t=0 18. for mi=0, mi<ms_1 19. if ms_1(mi) != ms_2(mi) 20. t++ 21.end for 22. End if 23.End function

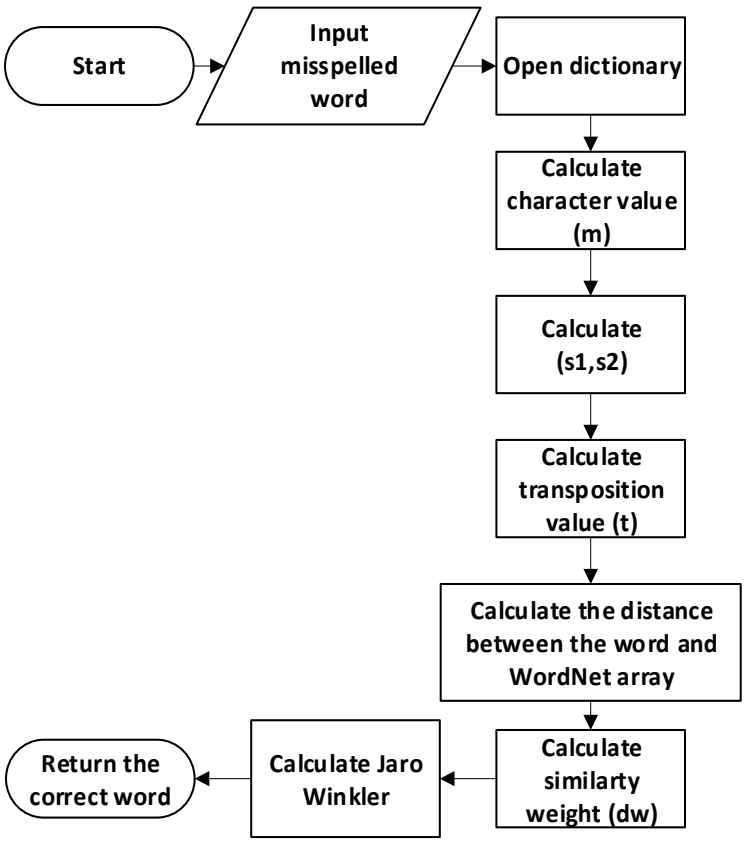

Figure 7: Flowchart of Misspelling Process

5.3.3 Spell Checking

The following diagram states the process of spell-checking to check the correctness of the entered word, first its run the *spell checker* function which tokenize the word (convert it to numbers) and compared it to a dictionary of numbers (spellchecker

dictionary), then calculate the value between the entered word and the words in the dictionary. When the value equal to -1 that's mean the word spelled correctly otherwise it misspelled. After this stage, all data converted to lower-case and saved in ontology.

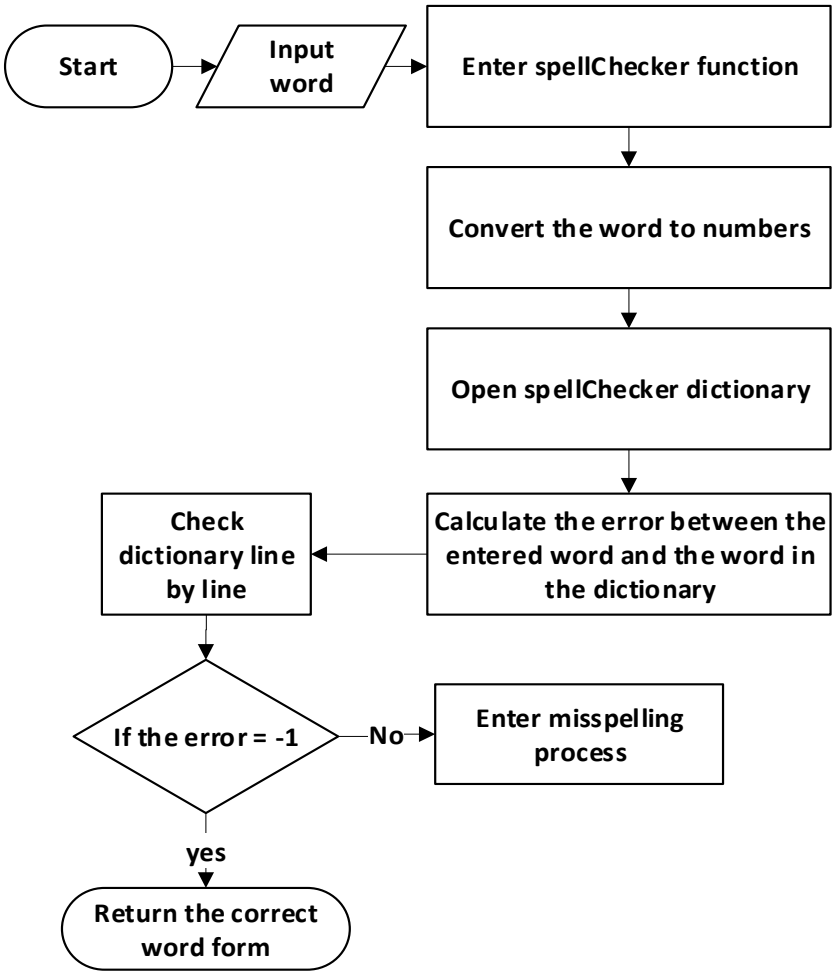

Figure 8: Flowchart of Spell Checking

Final step in pre-processing stage is synonym matching, where it matches the existing records inside ontology with input data using WordNet dictionary. Since the first days of NLP, Computational lexicons containing lists of words for computational processing have been generated, this list often labelled using POS [32]. Which categorize words according to their distribution in texts. Using WN a synset will be built, where every synonym set or synset contains a set of words in which all these words have the same meaning (synonymous).

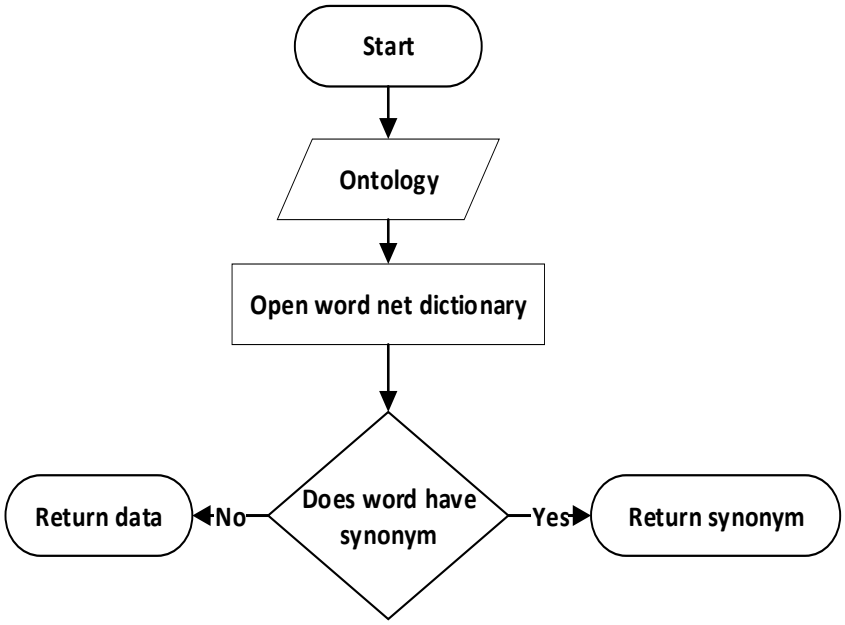

Figure 9: Flowchart of Synonym-Matching

5.4 Matching Stage

The process of matching words is done using cosine similarity, when the ontology is not empty it calculate the value between the input word and the one that already saved in ontology, if the value is more than 0.8 that's mean the two words are matching, then the matched values will be saved in an array and displayed as suggestion.

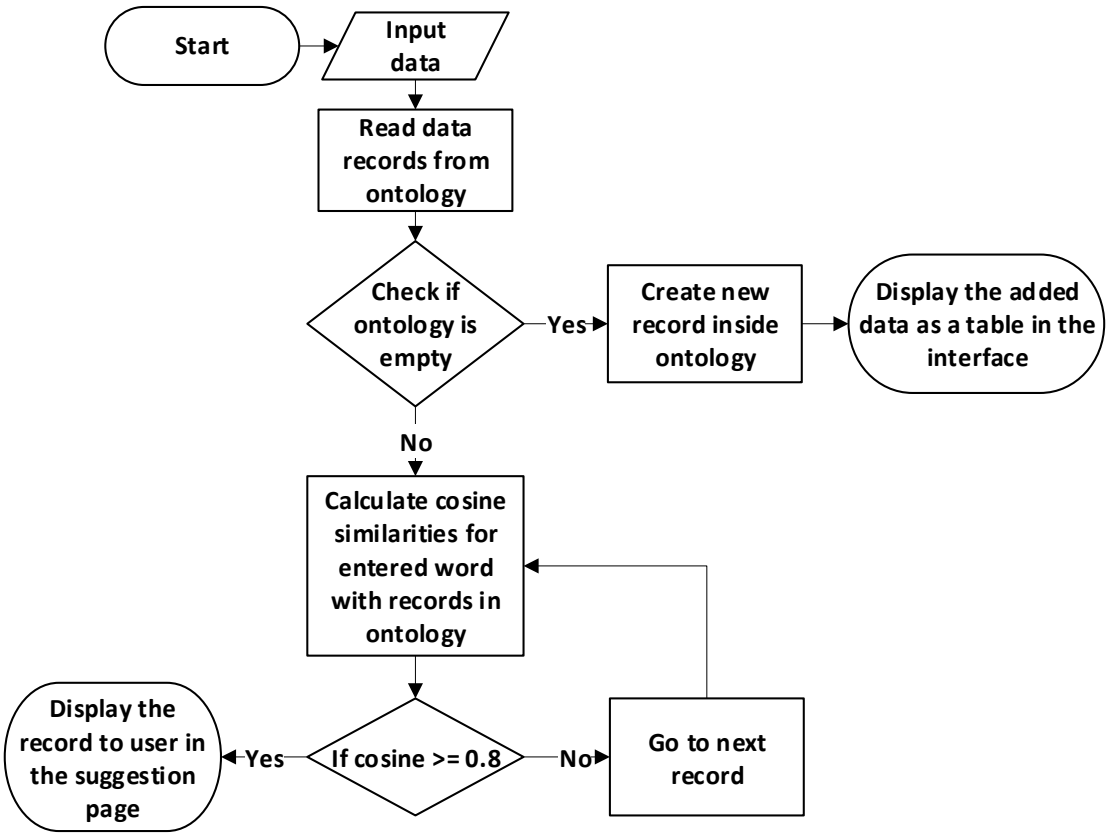

Figure 10: Flowchart of Matching Process

Otherwise, when the system is runs the ontology will be initialized then user start to enters data in the interface, both table and field box should have data, in the following flowchart displays the process of saving new record to ontology.

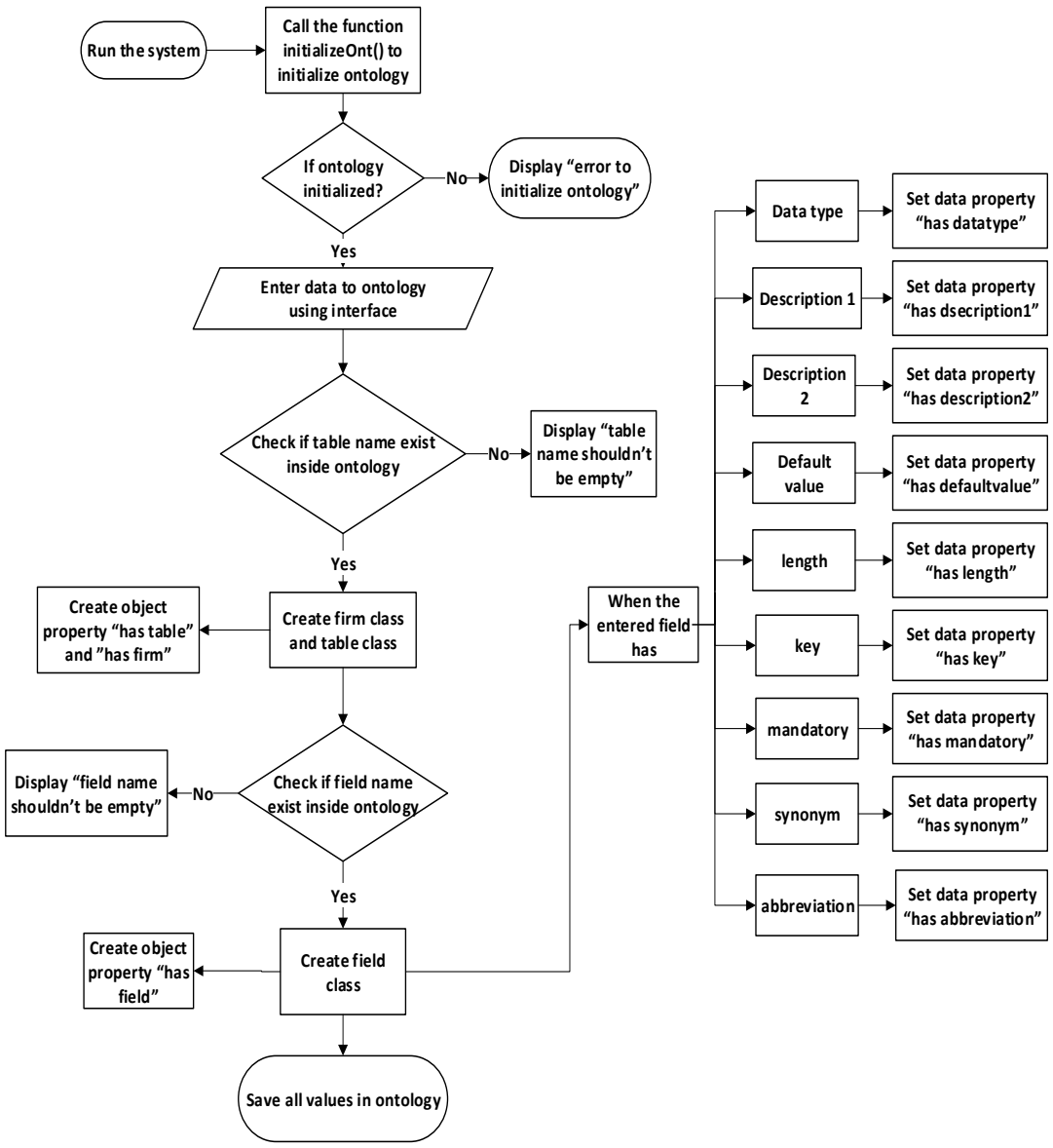

Figure 11: Flowchart of Saving New Record in Ontology

5.5 CNN Structure

The processes that applied on the train database, starts by encoding the trained data by using ASCII values, that converts the data into vectors (character wise). Then enters

the layers of the CNN structure. The result of last layer will be an array of clustered values, which will be saved in ontology as new class *schema*.

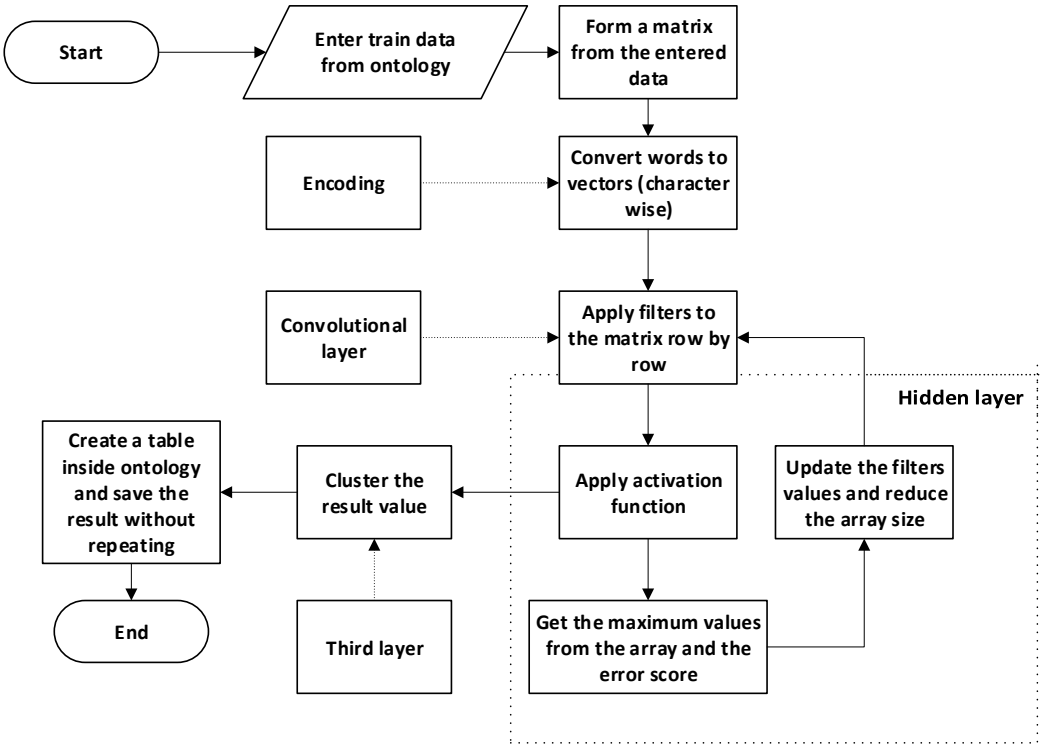

Figure 12: Flowchart of CNN Process

The clustering method working as following:

1. It creates a list of the first value taken from activation function; the index of this list will be one.
2. For the next value, it checks if the available list has a similar one, it will be added to this list.
3. Otherwise, if the next value is not similar it creates a new list and save it there.
4. This procedure will continue for all the value that obtained from activation function.
5. These clustered lists are the result of the proposed system, which will be saved in ontology as class schema.

Chapter 6

EXPERIMENTAL METHODOLOGY

6.1 Introduction

In this chapter, thesis methodology is explained, problem statements, explanations of the data that have been used, the structure of ontology system, the structure of CNN clustering and the structure of schema matching.

6.2 Problem Statement

The aim of this thesis is to promote a matching system that involves required steps to produce a map while matching relational databases to ontology representation using CNN's machine algorithm according to the previous article that used CNN gets impressive results in the field of schema matching and text classifying. The database field names used to test the algorithm, where it converted each field names of the 302 to vectors and performed a matrix to be processed later in the layers of CNN. The results of this algorithm compared with a list of expected fields, which is available in the database.

6.3 Data Collection Description

The database that used in this system was an e-commerce database of four companies (N11, Mikro, Logo, and Serotonin). This database consists of 302 fields and four table each firm has one (product, stock, item, products) and each table has many fields and each field has properties as (name, discerption1, discerption2, datatype, length, default value, key). For each field name there is an expected field name used as testing database. The database has two parts, table name and field name that entered by the

user and processed by the system that used as training dataset, and the expected field name that used as testing dataset.

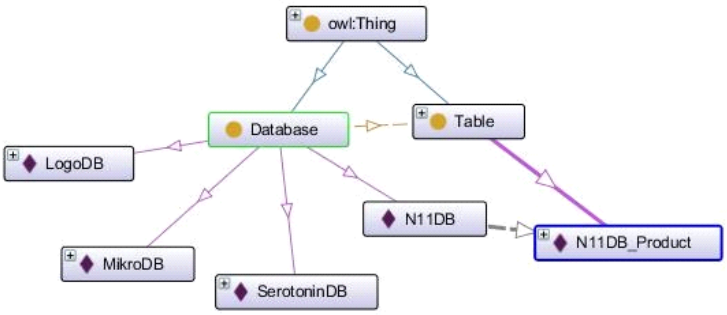

Figure 13: Database Schema

6.4 Ontology System Structure

An inference engine built which reads all the data from user-interface and stores it in ontology, there are five classes, class for every firm and one class will be initialize for the result of the CNN clustering, and a sub-class" table" to each Class. There are three restrictions: table name, field name and data type can't be empty and should have a value, also there are 7 object properties: (hasType, hasKey, hasDescription1, hasDescription2, hasMandatory, hasDefaultValue, hasLength). In addition, it has two relations for example; when the user entered a word, which has a synonym it will display in the interface relation's box as "has Synonym+ word", and when the entered word has an abbreviation, it will display in the interface relation's box as "has Abbreviation + abbreviation-word".

6.5 CNN Clustering Structure

As mentioned before the data needs to be encoded before it enters the CNN layers, where encoding is the process of transforming a text data in to a binary number so the computer can understand it, ASCII encoding used in the system to convert the words to vectors. It converts each letter to a number then these numbers will combine and

perform a vector for each data word. After that, the encoded data will enter the CNN layers, which consists of a convolutional layer, a hidden layer and an output layer. In convolutional layer, the encoded fields name will be the input in this layer, the number of hidden units in the neural network will affect the generalization output, in the system the number of units set to 400. The training matrix, which contain the field names entered to the convolutional layer, and then it multiplied with a filter by dot product, and it done row by row, the filters chosen by taking the longest field names. In the hidden layer, the sigmoid function was applied which produced positive numbers between the range [0,1]. It is the most useful function comparing with the other activation functions for training data that is also between 0 and 1. The result of this layer is an array of values between [0, 1]. Sigmoid function formula:

$$Sigmoid(y) = \frac{1}{e^{-x}} \tag{5}$$

In the third layer, the resulted array will be the input of this layer and the maximum value, which is closed to the defined threshold will be chosen, a backpropagation [33] algorithm used to update the difference of weights between each layer and update the filters and thresholds.

6.6 Schema Matching Structure

When the system runs, the ontology engine will be initialized by the function intializeOnt() and it will start to read the data that being entered by user using user interface. If the user did not enter “table name, field name or data type values”, an alarm massage will pop to the screen for user to enter the empty fields. The entered data will be processed to be understandable for the system. After the pre-process stage that described before, the data will be ready to enter the ontology. First the system will check if ontology is empty, the word will be saved with its object proprieties, if it’s not empty, the system will check, if the entered word matches to another word already

saved in ontology it will shows it as a suggestion list for the user whether to select or not. This matching stage happens in one table using similarity function and WordNet to get the synonym match.

The system will get the list of matched fields from ontology, and use it as training data, which will be encoded to vectors, the training data will be extracted and entered to the convolutional layer when a product dot is applied between the extracted array and the filter, which will form new array. The result of the first layer will enter to the activation function, which in his turn will performed an array of a range between [0, 1].

The output of it will be a table of matched fields and it will be saved separately inside the ontology as clustered matched table. An array with multiple values for each word will be the entry of the next layer, in the last layer the maximum value that close to the defined threshold will selected. If two words have the same value or similar one it will be saved in the same group, otherwise it will be saved in a new group. That all happened with the use of the backpropagation algorithm to calculate the differences of weights between the layers to update the filters and thresholds. Therefore, the result of this system is a group of clusters that have values of similar values in each cluster without any repetitions; finally, this list of clustered matching data will be saved in to ontology.

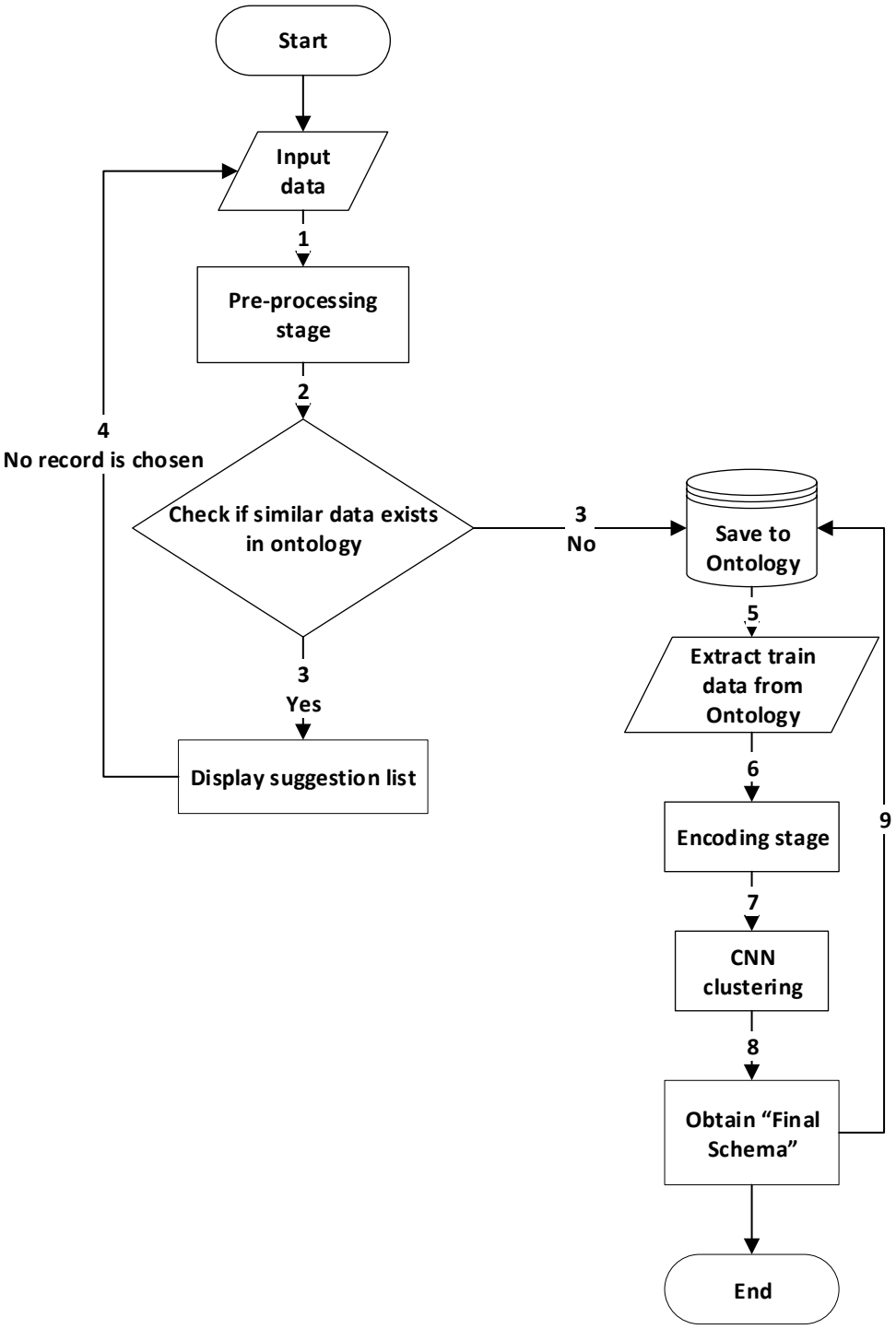

Figure 14: Schema Matching Process Flowchart

Chapter 7

SYSTEM INTERFACES

7.1 Introduction

In this chapter, interfaces of the applied system will be explained and displayed with screenshots including case studies with the used database, NetBeans and protégé used to build the proposed system.

7.2 Main Page Interface

When the system will run, the main interface will be displayed at first with the showing text boxes, which let the user to enter database. For the table name it is consisting of two parts separated by underscore, the first one is the firm name and the other one is the table name. Field name will be entered as well, then a combo box displays the data types where the user can choose from, then the length of the entered data will be next, if there is a default value of the entered field, it will be entered in the text box. If the entered filed is mandatory or not, it takes two values 1 for positive and 0 for negative. If the field is a primary key it's entered as PK, if it's a foreign key it's entered as FK if either it's kept empty. Finally, when the field had a description the description box will be filled as displayed below, as mentioned before table name, field name and data type should not be empty, all data will be saved in ontology as a new record.

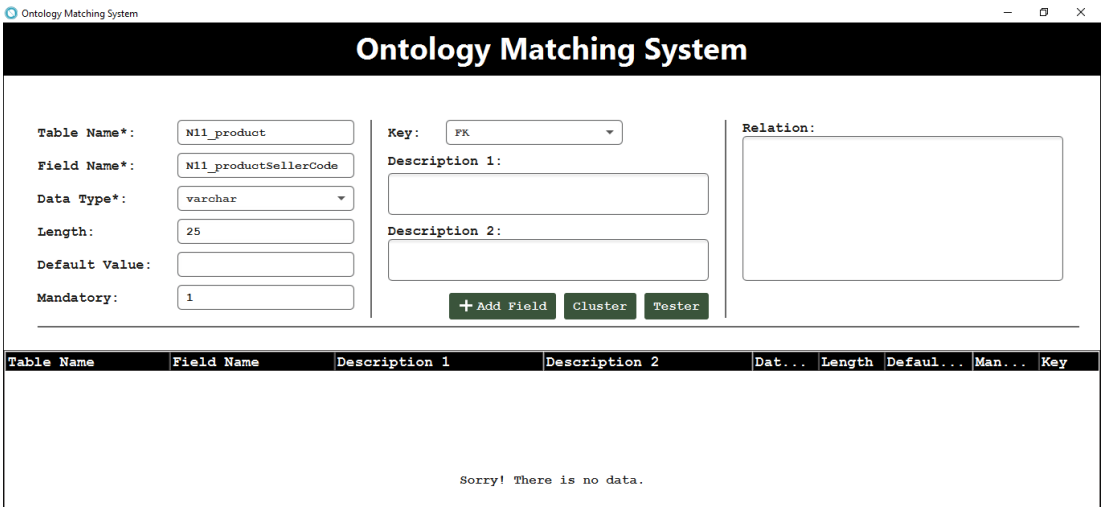

Figure 15: Main Page Interface

The user entered “N11_product” as table name, so the firm name “N11” will be saved in class firm with its title, and table class “product” will be created. The field name that entered by user “product seller code” the product word will be deleted and saved in ontology as” seller_code” as shown below.

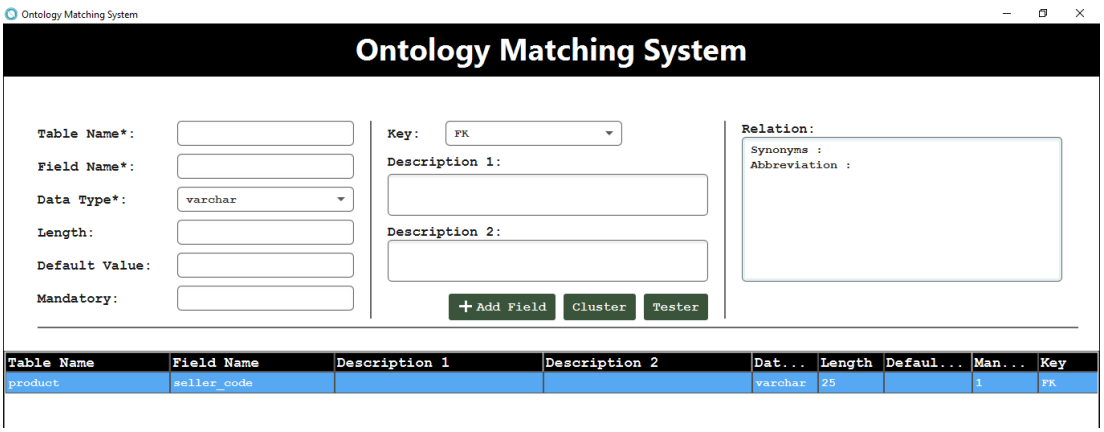

Figure 16: First Field Entry

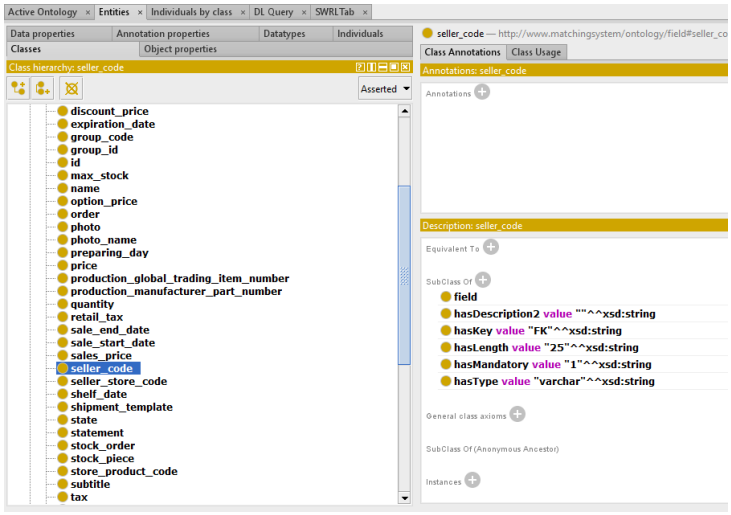

Figure 17: Screenshot of Seller Code Field Inside Ontology

7.3 Suggestion Interface

When user enters a field for the second time, system will check for the synonym similarities between the records that already saved in ontology and the data that entered to interface. Using WordNet and word similarity, the synonym score calculated. In the following figure when a user from another firm enters “currency type” which is already saved in ontology, it will be displayed as suggestion.

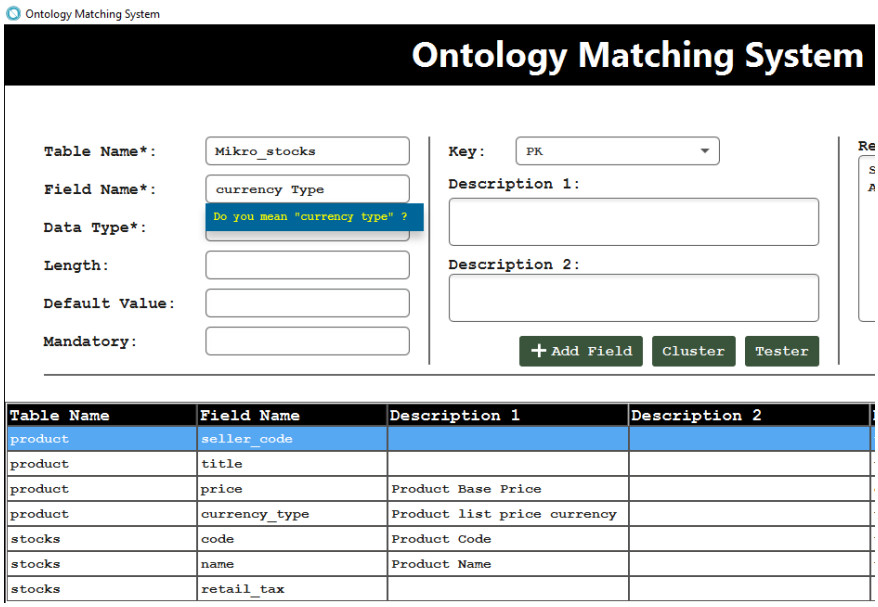

Figure 18: Screenshot Showing Suggestion

When a user of the same firm enters the same name field twice, suggestion page will be displayed for matching the two field and not to have repeating fields in same table.

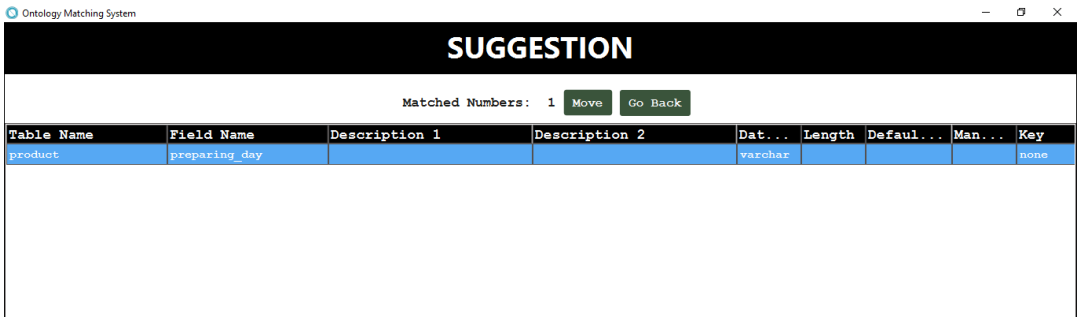

Figure 19: Matching Two Field in Same Table

7.4 Extraction Abbreviation Interface

In the following figure the abbreviation process shown, when the user enters a field VAT, it will be processed using abbreviation dictionary, and displayed in the relation box that, the entered field have abbreviation relation. The word will convert to lower case and saved in ontology with its full form value added tax.

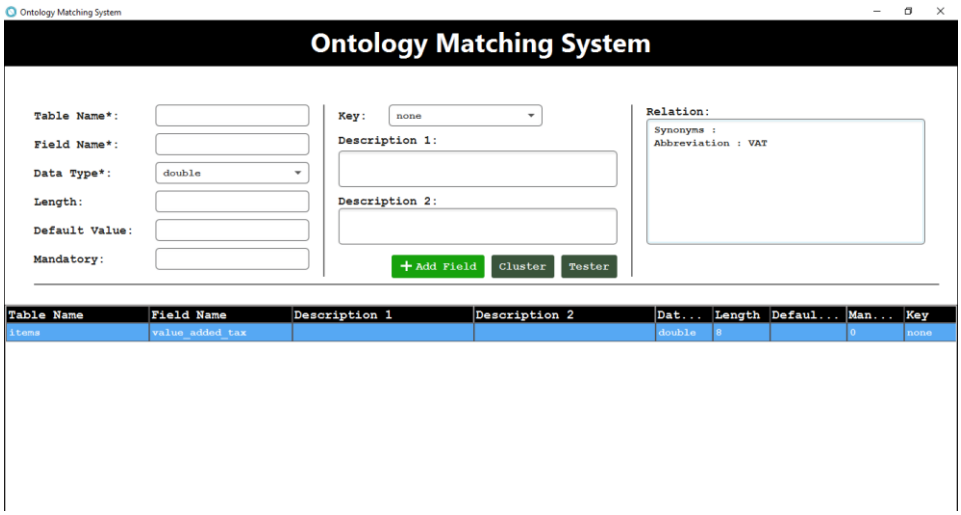

Figure 20: Extraction Abbreviation Interface

7.5 Clustering List Interface

When all database entered and processed in to ontology, the trained data encoded and enters CNN layers to match the fields between different tables, when user press on the cluster button all similar values will be listed in one group. As showing in the following figure “seller code, stock code, code, seller store code ….” all the fields that have the word “code” listed in one cluster, the result of clustering will be saved in ontology as schema class with the number of clustering; this process takes long time to be processed.

Table Name	Field Name	Dat...	Length	Default ...	Man...	Key	Clust
product	seller_code	varchar	25		1	PK	1
product	seller_store_code	varchar	30		0	none	1
stocks	stock_code	varchar	25		1	PK	1
stocks	store_product_code	varchar	25			none	1
stocks	producer_code	varchar	25		0	none	1
items	code	varchar	25		1	PK	1
items	group_code	varchar	17		0	none	1
items	producer_code	varchar	25		0	none	1
items	spec_code	varchar	11		0	none	1
items	authority_code	varchar	11		0	none	1
items	dominant_code	varchar	25		0	none	1
items	class_code	varchar	25		0	none	1
products	product_code	varchar	35		1	PK	1
product	title	varchar	35		1	none	2
product	price	double	8		1	none	3
product	option_price	varchar	25		0	none	3
items	price	double	8		1	none	3
products	sales_price	varchar	8		0	none	3
products	discount_price	float	10		0	none	3
product	currency_type	varchar	30		1	none	4

Figure 21: Clustering List Table

Chapter 8

EXPERIMENTAL RESULTS AND EVALUATION

8.1 Introduction

This chapter displays the experimental results and the evaluation of the methods used in this study. Four databases used in this thesis for testing (N11, MIKRO, SEROTONIN, and LOGO). 302 field names are used as mentioned before just one table is used from each database. From N11 database the table “product” is used which contain 25-field names, from MIKRO database the table “stocks” is used which contain 198-field names, from SEROTONIN database the table “products” is used which contain 15-field names and from LOGO database the table “items” is used which contain 64-field names. Two type of evaluation are performing, the first one done manually by comparing database tables with user inputs, the result of this evaluation performed an “Expert Schema”. The other evaluation done by comparing the system final schema with the Expert schema. To find the results of the evaluations Precision, Recall, F1-score and Accuracy calculated by the using of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) matrixs.

8.2 Evaluation 1: Comparing Database Tables with User Inputs

Precision, Recall, F-measure applied to get the percentage of accuracy, which it based on the confusion metric (true positives, false positive, true negative and false negative). To calculate these values an evaluation between the database tables and user input tables is obtained.

Where TP indicates the number of resulted words, which really matches the expected ones (exact match) or its synonyms, FP indicates the number of resulted words, which match the synonym of the entered ones with no wrong suggestions, TN indicates the number of resulted words which not matching the expected and have wrong suggestions and FN indicates the number of resulted words, which not exist in the expected data, with no correct suggestions.

For example, the user table contains the field names (product code, photo name, product name, sales price, currency type, order, statement, order day, group id, discount price, stock piece, tax, state, production date, size details).

If the user enters the input “Item name” the system will display the suggestions of the synonyms or exact term if exist in the system database as “product name”, “item name”, “stock name” and etc., then this considers as true positive. If the system displays the suggestions with wrong terms as “name”, “id”, “code” etc., then this considers as true negative. If the system did not give any suggestion with wrong terms then this considers as false positive. If the system did not suggest any correct terms as “group id” this considers as false negative which mean mismatch.

The following table shows the values of the confusion matrix, which is useful to calculate precision, recall and f1-score to find the accuracy of this methodology.

Table 7: Confusion Matrix 1

		Actual	
Predicted		True (T)	False (F)
	Positive (P)	208	16
	Negative (N)	66	12

By applying, the formula of accuracy can find how accurate this study is.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

Precision calculates the percentage of the results, which is relevant to the actual data.

$$precision = \frac{TP}{TP + FP} \quad (7)$$

Recall refers to the percentage of the results, which is correctly matched the actual data.

$$recall = \frac{TP}{TP + FN} \quad (8)$$

Precision and recall are used to compute the F1-score to finds the accuracy of the test using this formula.

$$f1 - score = 2 \times \frac{recall \times precision}{recall + precision} \quad (9)$$

After obtaining the values from confusion matrix and calculating the previous formulas the following table is established.

Table 8: Testing Results of First Evaluation

Accuracy	0.907
Precision	0.928
Recall	0.945
F1-score	0.936

8.3 Evaluation 2: Comparing System Final Schema with Expert Schema

Final schema it is the result that obtained from the proposed system, which is saved in the system database after the clustering stage as a final result, and the schema titled as "system final schema". In this evaluation the result of the system final schema will be

compared with the “expert schema” that obtained manually. The total number of fields are 302 in the first evaluation decreased to 239 fields because there wasn’t any repetition of the synonyms, while in the “system final schema” the fields decreased to 249 field names. The values of confusion metric are calculated after the comparison between the two schemas.

Table 9: Confusion Matrix 2

		Actual	
Predicted		True (T)	False (F)
	Positive (P)	159	19
	Negative (N)	50	11

After getting the values of (TP, TN, FP, FN) the accuracy of the second evaluation will be calculated by calculating the values of precision, recall and f1-score, so the results will be as following.

Table 10: Testing Results 2

Accuracy	0.874
Precision	0.893
Recall	0. 935
F1-score	0.913

8.4 Comparing Our Proposed Approach to Other Systems

In this section, a comparison will be occurred between this thesis study “schema matching using CNN clustering” with other two article studies related to the same subject but using different methods. More details will be explained next.

8.4.1 Comparing to Instance Based Schema Matching with Google Similarity and Regular Expression

A research done by M. A. Osama et al., "An Approach for Instance Based Schema Matching with Google Similarity and Regular Expression" [34], to solve the problem related to schema matching based on ontology (relation database), proposed an efficient schema matching approach to classify the similarity between attributes by the use of numeric instances, alphabet instances and deferent data types.

This approach uses the pattern recognition principle to establish instance-based regular expressions to define the matched attributes for numeric and deferent datatypes. In addition, for the alphabet data types, in this approach Google similarity was used to calculate the semantic similarity score in order to get the semantic relations between instances. The result of this approach comparing to our study is showing below.

Table 11: Results of Comparing Our Study with Google Similarity

	Our work	**Google similarity**
Precision	92.8%	96%
Recall	94.5%	93%
F1-score	93.6%	95%

In terms of the values of precision, recall and f1-score, the performance of the M. A. Osama et al, [34] gives better matching results comparing with our proposed approach because of the usage of Google similarity to find the sematic relation between terms.

8.4.2 Comparing to Schema Matching for Large-Scale Data Based on Ontology Clustering Method

Alani, H, "Schema Matching for Large-Scale Data Based on Ontology Clustering Method" [35], proposed an ontology-based clustering, the ontology that been used in

this research have rich semantic related to the field of schema matching. Alani, H used term frequency (TF-IDF) to generate the most frequent terms, noun compound extraction uses it by utilize a part-of-speech tagging (POS) and n-gram similarity, also rule-based clustering had performed with multiple algorithms for measuring the similarities including Dice, Cosine and Jaccard.

Table 12: Results of Comparing Our Study with Ontology Based Schema using cosine similarity

	Our work	**Ontology Based Schema(cosine)**
Precision	92.8%	95%
Recall	94.5%	90%
F1-score	93.6%	92%

Table 12, shows the experiment results of applying the Cosine similarity on ontology-based cluster comparing to our system where cosine similarity used as well to calculate the matched fields, our system performance gives better results because of the usage of WordNet dictionary and clustering using CNN algorithm.

Table 13: Results of Comparing Our Study with Ontology Based Schema using Jaccard

	Our work	**Ontology Based Schema (Jaccard)**
Precision	92.8%	88%
Recall	94.5%	85%
F1-score	93.6%	86%

Table 13, shows the experiment results of applying the Jaccard method on ontology-based cluster comparing to our system, in terms of the values of precision, recall and f1-score our proposed approach achieved better results from the performance of Alani, H [35].

Chapter 9

CONCLUSION

Fifteen years ago, it was really difficult for any Natural Language Processing business when dealing with the customers to express that the product they produced doesn't work with absolute precision, especially with the multiple datatypes in the blogs, social networks, e-commerce. Etc. This changed with time; NLP is a new field, full of promises and with an international community that continues to develop new algorithms, techniques, applications and tools. It moves fast. The recent results using deep learning are amazingly improving several hard NLP tasks. This study indicates a schema matching system that involves required steps to produce a map while matching relational databases to ontology with the use of CNN and cosine similarity. The combination of semantic schema matching with machine learning algorithm like CNN provide great improvement in the matching field, which become easier and time consuming. Using deep learning algorithms huge numbers of sellers and millions of products could be grouped inside the e-commerce sites, where could be more efficient and helps the customers to find what they looking for in a short time, the proposed system could be used as well with e-commerce systems to collaborate sharing business processes through the system. For future work, enhance the using of ontology by adding more relations and restrictions between the attributes and adding more layers to CNN structure to get more specific and accurate matching results.

REFERENCES

[1] Islam, A., & Inkpen, D. (2008). Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *2*(2), 1-25.

[2] Karasneh Y., Ibrahim H., Othman M., Yaakob R. (2010). Challenges in Matching Heterogeneous Relational Databases Schemas, IKE'10 - 9th International Conference on Information and Knowledge Engineering – USA.

[3] Bergamaschi, S., Beneventano, D., Po, L., Sorrentino, S. (2011). Automatic Normalization and Annotation for Discovering Semantic Mappings, Search Computing II, LNCS 6585, pp. 85–100, Springer.

[4] Rahm, E., Bernstein, P. A. (2001). A survey of approaches to automatic schema matching, The VLDBJournal 10: 334–350.

[5] Kim B., Namkoong H., Lee D., Hyun S. J. (2011). A Clustering Based Schema Matching Scheme for Improving Matching Correctness of Web Service Interfaces, International Conference on Services Computing, IEEE.

[6] Yang Y., Chen M., Gao B. (2008). An Effective Content-based Schema Matching Algorithm, International Seminar on Future Information Technology and Management Engineering, IEEE.

[7] Levenshtein, V. I. (1966, February). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady* (Vol. 10, No. 8, pp. 707-710).

[8] Zobel, J., & Dart, P. (1996, August). Phonetic string matching: Lessons from information retrieval. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 166-172).

[9] Do H-H, Rahm E. (2002). COMA - A system for flexible combination of schema matching approaches, Proceedings of the 28th VLDB Conference, Hong Kong, China.

[10] McGuinness, D. L., & Van Harmelen, F. (2004). OWL web ontology language overview. *W3C recommendation*, *10*(10), 2004.

[11] Wang, R. F. (2018). Semantic Text Matching Using Convolutional Neural Networks.

[12] Brownlee, J. (2019, September 25). How Do Convolutional Layers Work in Deep Learning Neural Networks? Retrieved from https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/.

[13] Pavel Shvaiko, Jerome Euzenat. (2005). A Survey of Schema-Based Matching Approaches In Journal on Data Semantics, pp.146-171.

[14] Kettouch, M. S., Luca, C., Hobbs, M., & Dascalu, S. (2017, September). Using semantic similarity for schema matching of semi-structured and linked data. In *2017 Internet Technologies and Applications (ITA)* (pp. 128-133). IEEE.

[15] Chen, L. J. (2016, July). Deep Web Schema Matching Based on Concept-Word and Semantic-Heterogeneous Model. In *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)* (pp. 289-293). IEEE.

[16] Jian, N., Hu, W., Cheng, G., Qu. Y. (2010). FalconAO: Aligning Ontologies with Falcon, Department of Computer Science and Engineering Southeast University.

[17] Madhavan, J., Bernstein, P. A., Rahm, E. (2001). Generic Schema Matching with Cupid. In Apers, P. M. G., Atzeni, P., Ceri, S., Paraboschi, S., Ramamohanarao, K., and Snodgrass, R. T., editors, Proc. of the 27th International Conference on Very Large Data Bases (VLDB 2001), September 11-14, 2001, Roma, Italy, pages 49–58. Morgan Kaufmann.

[18] Sergey Melnik ,H. Garcia-Molina,Erhard Rahm (2002)Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching ,In: Proceedings 18th International Conference on Data Engineering.

[19] Sorrentino, S., Bergamaschi, S., & Gawinecki, M. (2011, April). NORMS: an automatic tool to perform schema label normalization. In *2011 IEEE 27th International Conference on Data Engineering* (pp. 1344-1347). IEEE.

[20] Doan, A., Madhavan, J., Domingos, P., & Halevy, A. (2002, May). Learning to map between ontologies on the semantic web. In *Proceedings of the 11th international conference on World Wide Web* (pp. 662-673).

[21] Li, W. S., & Clifton, C. (2000). SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. *Data & Knowledge Engineering*, *33*(1), 49-84.

[22] Dos Santos, C., & Gatti, M. (2014, August). Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (pp. 69-78).

[23] Wang, P., Xu, B., Xu, J., Tian, G., Liu, C. L., & Hao, H. (2016). Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. *Neurocomputing*, *174*, 806-814.

[24] Rezaeinia, S. M., Ghodsi, A., & Rahmani, R. (2018). Text Classification based on Multiple Block Convolutional Highways. *arXiv preprint arXiv:1807.09602*.

[25] Zhang, J., Liu, F. A., Xu, W., & Yu, H. (2019). Feature Fusion Text Classification Model Combining CNN and BiGRU with Multi-Attention Mechanism. *Future Internet*, *11*(11), 237.

[26] Salehinejad, H., Barfett, J., Aarabi, P., Valaee, S., Colak, E., Gray, B., & Dowdell, T. (2017, October). A convolutional neural network for search term detection.

In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)* (pp. 1-6). IEEE.

[27] Majumder, N., Poria, S., Gelbukh, A., & Cambria, E. (2017). Deep learning-based document modeling for personality detection from text. *IEEE Intelligent Systems*, *32*(2), 74-79.

[28] N11.com - Alışverişin Uğurlu Adresi. (2020). Retrieved 6 February 2020, from https://www.n11.com/.

[29] ERK TEKNOLOJİ. (2020). Retrieved 25 February 2020, from http://www.erkteknoloji.com/.

[30] Malzemeler Veri Aktarımları - Tiger 3 Bilgi Deposu - Global Site. (2020). Retrieved 4 February 2020, from https://docs.logo.com.tr/pages/viewpage.action?pageId=22272929.

[31] Mikro Tablolar. (2020). Retrieved 4 February 2020, from http://www.mye.com.tr/help/Library/Diger/DBYapisi_V15/tablo.html.

[32] Taljard, E. E., Faab, G., & Bosch, S. (2015). Implementation of a part-of-speech ontology: morphemic units of Bantu languages.

[33] Ramasundaram, S., & Victor, S. P. (2010). Text categorization by backpropagation network. *International Journal of Computer Applications*, *8*(6), 1-5.

[34] M. A. Osama, I. Hamidah, and A. S. Lilly, “An approach for instance based schema matching with Google similarity and regular expression,” The Int. Arab J. of Info. Tech. Pp. 755- 763, 2017.

[35] Alani, H., & Saad, S. (2017). Schema matching for largescale data based on ontology clustering method. *International Journal on Advanced Science, Engineering and Information Technology*, *7*(5), 1790-1797.

APPENDICES

Appendix A: Database Table

TABLE NAME	FIELD NAME	FIELD DESCRIPTION 1	DATA TYPE	LENGTH	DEFAULT VALUE	MANDATORY (1/0)	KEY
N11_product	N11_productSellerCode		varchar	25		1	FK
Mikro_stocks	Mikro_stock_code	Product Code	varchar	25			PK
LG_ITEMS	LG_CODE	Material Code	varchar	25		1	
Serotonin_product	SER_productCode		varchar	255		0	
N11_product	N11_title		varchar	35		1	
Mikro_stocks	Mikro_stock_name	Product Name	varchar	50			
LG_ITEMS	LG_NAME	Material Description	varchar	51		0	
Serotonin_product	SER_name		varchar	255		1	
N11_product	N11_price	Product Base Price	double			1	
LG_ITEMS	LG_STCOMPLN.PRICE	Price	double	8		0	
Serotonin_product	SER_sales_price		varchar	20		0	
Mikro_stocks	Mikro_stock_retailTax	Retail Tax Rate	Tinyint				
LG_ITEMS	LG_VAT	tax	double	8		0	
Serotonin_product	SER_tax		float		18	0	
N11_product	N11_currencyType	Product list price currency	varchar	30		1	
Mikro_stocks	Mikro_stock_currencyType	Currency Id	Tinyint				
Serotonin_product	SER_currencyType		varchar	5	1	0	
N11_product	N11_url	Product official image URL	varchar			1	
LG_ITEMS	LG_IMAGE	Picture	byte	1		0	
Serotonin_product	SER_photoName		varchar	255		0	
N11_product	N11_order	Product image display order	varchar			1	
Serotonin_product	SER_order		int	11		0	
N11_product	N11_productCondition	Product State	varchar			1	
LG_ITEMS	LG_ACTIVE	Registration Status	int	2		1	
Serotonin_product	SER_state		varchar	2	1	0	
N11_product	N11_discount	Product discount information	varchar			1	
Serotonin_product	SER_discountPrice		float	10.2		0	

N11_product	N11_quantity	The amount of product	varchar			1	
Mikro_stocks	Mikro_sto_maxStok		float				
Serotonin_product	SER_Stock Piece		int	11		0	
N11_product	N11_subtitle		varchar			1	
N11_product	N11_description	Product description information (can be HTML)	varchar			1	
Serotonin_product	SER_statement		text			0	
N11_product	N11_attribute	name and value of the Product properties are entered in the field	varchar			0	
N11_product	N11_id	Product category number	varchar			1	
Serotonin_product	SER_GroupIdd		varchar	20		0	
N11_product	N11_saleStartDate	Product sales start date (dd / MM / yyyy)	varchar			0	
N11_product	N11_saleEndDate	Product sales end date (dd / MM / yyyy)	varchar			0	
N11_product	N11_productionDate	Product Production Date (dd / mm / yyyy)	varchar			0	
N11_product	N11_expirationDate	Product expiration date (dd / MM / yyyy)	varchar			0	
LG_ITEMS	LG_SHELF DATE	Expiration date	int	2		0	
Serotonin_products	SER_expirationDate		varchar	50		0	
N11_product	N11_preparingDay	Production Time to ship (in days)	varchar			1	
Mikro_stocks	Mikro_stock_order	Order time (days)	int				
N11_product	N11_shipmentTemplate	Delivery Template Name				1	
N11_product	N11_sellerStoreCode	Store Product code	varchar			0	
Mikro_stocks	Mikro_sto_productCode	Store Product product Code	varchar	25			
N11_product	N11_optionPrice	List price of the product product unit	varchar			0	
N11_product	N11_bundle	Bundle Products	varchar			0	

N11_product	N11_mpn	Production Manufacturer Part Number	varchar			0	
N11_product	N11_gtin	Production global trading item number	varchar			0	
LG_ITEMS	LG_CARDTYPE	Material Registration Type	int	2		1	
LG_ITEMS	LG_STGRPCODE	Material Group Code	varchar	17		0	
N11_product	N11_gcin	Production global commercial item number	varchar			0	
Mikro_stocks	Mikro_sto_producerCode	Manufacturer Code	varchar	25			
LG_ITEMS	LG_PRODUCERCODE	Manufacturer Code	varchar	25		0	
LG_ITEMS	LG_SPECODE	Special code	varchar	11		0	
LG_ITEMS	LG_AuthorityCODE	Authorization Code	varchar	11		0	
LG_ITEMS	LG_CLASSTYPE	Material Class Type	int	2		0	
LG_ITEMS	LG_PURCHBRWS	Place of Use - purchasing	int	2		0	
LG_ITEMS	LG_SALESBRWS	Place of Use - Sales and Distribution	int	2		0	
LG_ITEMS	LG_MTRLBRWS	Place of Use - Store Management	int	2		0	
LG_ITEMS	LG_TRACKTYPE	Track Type	byte	1		0	
LG_ITEMS	LG_LOCTRACKING	Product Inventory Tracking	byte	1		0	
LG_ITEMS	LG_TOOL	Tool	byte	1		0	
LG_ITEMS	LG_AUTOINCSL	Automatically Increase Product Serial Number	byte	1		0	
LG_ITEMS	LG_DIVLOTSIZE	Lot Size Can Be Split	byte	1		0	
Mikro_stocks	Mikro_sto_Shelf life	Shelf life	int				
LG_ITEMS	LG_SHELFLIFE	Shelf life	double	8		0	
LG_ITEMS	LG_DEPRTYPE	DepreciationType	int	2		0	
LG_ITEMS	LG_DEPRRATE	DepreciationRate	double	8		0	
LG_ITEMS	LG_DEPRDUR	DepreciationDuration	int	2		0	
LG_ITEMS	LG_SALVAGEVAL	SalvageValue	double	8		0	
LG_ITEMS	LG_REVALFLAG	Revaluation	byte	1		0	
LG_ITEMS	LG_REVDEPRFLAG	ValuationDepreciation	byte	1		0	

LG_ITEMS	LG_PARTDEP	Part Depreciation	int	1		0	
LG_ITEMS	LG_DEPRTYPE2	Depreciation Type2	double	2		0	
LG_ITEMS	LG_DEPRRATE2	Depreciation Rate2	int	8		0	
LG_ITEMS	LG_DEPRDUR2	Depreciation duration2	real	2		0	
LG_ITEMS	LG_REVALFLAG2	Revaluation 2	byte	1		0	
LG_ITEMS	LG_REVDEPRFLAG2	Alternative Valuation Depreciation	byte	1		0	
LG_ITEMS	LG_PARTDEP2	Part Depreciation2	byte	1		0	
LG_ITEMS	LG_APPROVED	Approved	byte	1		0	
LG_ITEMS	LG_DISTAMOUNT	DistributedAmount	double	8		0	
LG_ITEMS	LG_CAPIBLOCK_CREATEDBY	Created By	int	2		0	
LG_ITEMS	LG_CAPIBLOCK_CREADEDDATE	Created Date	longint	4		0	
LG_ITEMS	LG_CAPIBLOCK_CREATEDHOUR	Created hour	int	2		0	
LG_ITEMS	LG_CAPIBLOCK_CREATEDMIN	Created Minute	int	2		0	
LG_ITEMS	LG_CAPIBLOCK_CREATEDSEC	Created Secound	int	2		0	
LG_ITEMS	LG_CAPIBLOCK_MODIFIEDBY	Modifed	int	2		0	
LG_ITEMS	LG_CAPIBLOCK_MODIFIEDDATE	ModifedDate	longint	4		0	
Serotonin_product	SER_dateOfUpdate		varchar	30		0	
LG_ITEMS	LG_CAPIBLOCK_MODIFIEDHOUR	ModifedHour	int	2		0	
LG_ITEMS	LG_CAPIBLOCK_MODIFIEDMIN	ModifedMinute	int	2		0	
LG_ITEMS	LG_CAPIBLOCK_MODIFIEDSEC	Modifed Seconds	int	2		0	
LG_ITEMS	LG_SITEID	Data Center	int	2		0	
LG_ITEMS	LG_ORGLOGICREF	Data Reference	longint	4		0	
LG_ITEMS	LG_UNIVID	Out of use	varchar	25		0	
LG_ITEMS	LG_DISTLOTUNITS	Lot Size can be Distributed	byte	1		0	
LG_ITEMS	LG_COMBLOTUNITS	Lot Sizes Combineable	byte	1		0	
LG_ITEMS	LG_LOTSIZINGMTD	Lot Determination Method	int	2		0	

LG_ITEMS	LG_FIXEDLOTSIZE	Fixed Lot Size	double	8		0	
LG_ITEMS	LG_YIELD	Yield	double	8		0	
LG_ITEMS	LG_MINORDERQTY	Minimum Order Quantity	double	8		0	
LG_ITEMS	LG_MAXORDERQTY	Max Order Quantity	double	8		0	
LG_ITEMS	LG_MULTORDERQTY	Multi Order Quantity	double	8		0	
LG_ITEMS	LG_DOMINANTCODE	MaterialCard Code	varchar	25		0	
LG_ITEMS	LG_CLASSTYPE	Class Type	int	2		0	
LG_ITEMS	LG_CLASS_CODE	Class Code	varchar	25		0	
LG_ITEMS	LG_DISTPOINT	Distribution Point	double	8		0	
LG_ITEMS	LG_CANUSEINTRNS	Used in Movements (1: True 2: False)	byte	1		0	
Mikro_stocks	Mikro_sto_RECno		Integer IDENTITY				
Mikro_stocks	Mikro_sto_RECid_DBCno		Smallint				
Mikro_stocks	Mikro_sto_RECid_RECno		Integer				
Mikro_stocks	Mikro_sto_SpecRECno		Integer				
Mikro_stocks	Mikro_sto_cancel		Bit				
Mikro_stocks	Mikro_sto_fileid		Smallint				
Mikro_stocks	Mikro_sto_hidden		Bit				
Mikro_stocks	Mikro_sto_locked		Bit				
Mikro_stocks	Mikro_sto_changed		Bit				
Mikro_stocks	Mikro_sto_checksum		Integer				
Mikro_stocks	Mikro_sto_create_user		Smallint				
Mikro_stocks	Mikro_sto_create_date		DateTime				
Mikro_stocks	Mikro_sto_lastup_user		Smallint				
Mikro_stocks	Mikro_sto_lastup_date		DateTime				
Mikro_stocks	Mikro_sto_special1		varchar	4			
Mikro_stocks	Mikro_sto_special2		varchar	4			
Mikro_stocks	Mikro_sto_special3		varchar	4			

Mikro_stocks	Mikro_sto_shortName	sto short name	varchar	25			
Mikro_stocks	Mikro_sto_foreignName	sto foreign name	varchar	50			
Mikro_stocks	Mikro_sto_SellerCurrentCode	Seller current code	varchar	25			
Mikro_stocks	Mikro_sto_type	Product Type	Tinyint				
Mikro_stocks	Mikro_sto_following Details	following Details	Tinyint				
Mikro_stocks	Mikro_sto_unit1_name	unit name	varchar	10			
Mikro_stocks	Mikro_sto_unit1_coefficient	unit1 coefficient	Float				
Mikro_stocks	Mikro_sto_unit1_weight	Unit Net Weight (kg)	Float				
Mikro_stocks	Mikro_sto_unit1_width	unit width (mm)	Float				
Mikro_stocks	Mikro_sto_unit1_length	unit length (mm)	Float				
Mikro_stocks	Mikro_sto_unit1_height	unit height (mm)	Float				
Mikro_stocks	Mikro_sto_unit1_tare	unit1 tare	Float				
Mikro_stocks	Mikro_sto_unit2Name	unit Name	varchar	10			
Mikro_stocks	Mikro_sto_unit2_coefficient	unit coefficient	Float				
Mikro_stocks	Mikro_sto_unit2_weight	unit Net weight (kg)	Float				
Mikro_stocks	Mikro_sto_unit2_width	unit width (mm)	Float				
Mikro_stocks	Mikro_sto_unit2_length	unit length (mm)	varchar	10			
Mikro_stocks	Mikro_sto_unit2_height	unit height (mm)	Float				
Mikro_stocks	Mikro_sto_unit2_tare		Float				
Mikro_stocks	Mikro_sto_unit3_name	unit Name	Float				
Mikro_stocks	Mikro_sto_unit3_coefficient	unit coefficient	Float				

Mikro_stocks	Mikro_sto_unit3_weight	unit Net weight (kg)	Float				
Mikro_stocks	Mikro_sto_unit3_width	unit width (mm)	Float				
Mikro_stocks	Mikro_sto_unit3_length	unit length (mm)	varchar	10			
Mikro_stocks	Mikro_sto_unit3_height	unit height (mm)	Float				
Mikro_stocks	Mikro_sto_unit3_tare		Float				
Mikro_stocks	Mikro_sto_unit4_name	unit Name	Float				
Mikro_stocks	Mikro_sto_unit4_coefficient	unit coefficient	Float				
Mikro_stocks	Mikro_sto_unit4_weight	unit Net weight (kg)	Float				
Mikro_stocks	Mikro_sto_unit4_width	unit width (mm)	Float				
Mikro_stocks	Mikro_sto_unit4_length	unit length (mm)	varchar	40			
Mikro_stocks	Mikro_sto_unit4_height	unit height (mm)	varchar	40			
Mikro_stocks	Mikro_sto_unit4_tara		Nvarchar(40)				
Mikro_stocks	Mikro_sto_Acc_code	product Acc Account Code	varchar	40			
LG_ITEMS	LG_GLACC_CODE	Acc Account Code	varchar	25		0	
Serotonin_product	SER_AccCode		varchar	255		0	
Mikro_stocks	Mikro_sto_Acc_Return_code	sto_Acc_Return_code	varchar	40			
Mikro_stocks	Mikro_sto_Acc_purchase_Acc_code	Stok Acc. purchasing code	varchar	40			
Mikro_stocks	Mikro_sto_Acc_satIadAccCode	Stok Acc. purchasing return code	varchar	40			
Mikro_stocks	Mikro_sto_Acc_discount_code	product Acc. Discount Code	varchar	40			
Mikro_stocks	Mikro_sto_Acc_purchasing_discount_code	product Acc. purchasing Discount Code	varchar	40			

Mikro_stocks	Mikro_sto_Acc_sales_costs_code	product Acc. Sales Costs Code	varchar	40			
Mikro_stocks	Mikro_sto_Acc_overseas_sales_code	product Acc. Overseas Sales Code	varchar	40			
Mikro_stocks	Mikro_sto_Acc_extra_charges_code	product Acc. Extra Charges Code	varchar	40			
Mikro_stocks	Mikro_sto_invastment_promo_Acc_code	Investment Promotion of Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_stores_sales_Acc_code	Stores Sales Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_sales_cost_stores_Acc_code	Cost Of Sales Between Stores of Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_bagortsatAccCode	Partners, Depending On Sales Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_bagortsatladAccCode	Return Sales to Affiliates Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_bagortsatIskAccCode	Depending On The Partner Sales Discount Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_diff_sales_price_Acc_code	The Difference In The Sale Price Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_cost_export_sales_Acc_code	The Cost Of Export Sales Acc. Code	Float				
Mikro_stocks	Mikro_sto_bagort_sales_cost_Acc_code	bagort Sales Cost Acc. Code	Float				
Mikro_stocks	Mikro_sto_zero_cost_sales_IFRSdifference_kod	Zero Paid Cost Of Sales Acc. Code	Float				
Mikro_stocks	Mikro_sto_profit	Profit Rate	Float				
Mikro_stocks	Mikro_sto_min_stock	The Minimum product Level	Tinyint				
Mikro_stocks	Mikro_sto_order_stock	product Order Level	Tinyint				

Mikro_stocks	Mikro_sto_max_stock	Maximum product Level	Smallint				
Mikro_stocks	Mikro_sto_given_order_unit	Given Order Unit	Tinyint				
Mikro_stocks	Mikro_sto_received_order_unit	Received Order Unit	Tinyint				
Mikro_stocks	Mikro_sto_order_time	Order Time (Days)	varchar	25			
Mikro_stocks	Mikro_sto_retail_rate	Retail Vat Rate	Tinyint				
Mikro_stocks	Mikro_sto_wholesale_rate	Wholesale Vat Rate	Tinyint				
Mikro_stocks	Mikro_sto_warehouse_code	Warehouse Address	Tinyint				
Mikro_stocks	Mikro_sto_electronic_label_type	Electronic Label Type	Tinyint				
Mikro_stocks	Mikro_sto_shelf_label	Shelf Label	Tinyint				
Mikro_stocks	Mikro_sto_label_account	Print a label?	Tinyint				
Mikro_stocks	Mikro_sto_sales_stops	Sales Stop?	Bit				
Mikro_stocks	Mikro_sto_stop_order	Stop Order?	Bit				
Mikro_stocks	Mikro_sto_stop_accepted_goods	Will you accept the goods?	Bit				
Mikro_stocks	Mikro_sto_accepted_day1	Goods acceptance day	Bit				
Mikro_stocks	Mikro_sto_accepted_day2	Goods acceptance day	Bit				
Mikro_stocks	Mikro_sto_accepted_day3	Goods acceptance day	Bit				
Mikro_stocks	Mikro_sto_accepted_day4	Goods acceptance day	Bit				
Mikro_stocks	Mikro_sto_accepted_day5	Goods acceptance day	Bit				
Mikro_stocks	Mikro_sto_accepted_day6	Goods acceptance day	Bit				
Mikro_stocks	Mikro_sto_accepted_day7	Goods acceptance day	Bit				

Mikro_stocks	Mikro_sto_orders_day1	Order Days	Bit				
Mikro_stocks	Mikro_sto_orders_day2	Order Days	Bit				
Mikro_stocks	Mikro_sto_orders_day3	Order Days	Bit				
Mikro_stocks	Mikro_sto_orders_day4	Order Days	Bit				
Mikro_stocks	Mikro_sto_orders_day5	Order Days	Bit				
Mikro_stocks	Mikro_sto_orders_day6	Order Days	Bit				
Mikro_stocks	Mikro_sto_orders_day7	Order Days	Smallint				
Mikro_stocks	Mikro_sto_discount_can't_done	Discounts Can't Be Done?	varchar	25			
Mikro_stocks	Mikro_sto_Inliquidation	Short-lived provisional all?	varchar	25			
Mikro_stocks	Mikro_sto_sub_group_no	sub group number	varchar	25			
Mikro_stocks	Mikro_sto_category_code	product Category Code	varchar	25			
Mikro_stocks	Mikro_sto_product_officier_code	Product Officer Code	varchar	25			
Mikro_stocks	Mikro_sto_invent_subgroup_code	Inventory Subgroup Code	varchar	25			
Mikro_stocks	Mikro_sto_parent_group_code	product Of The Parent Group Code	varchar	25			
Mikro_stocks	Mikro_sto_producer_code	Manufacturer Code	varchar	25			
Mikro_stocks	Mikro_sto_sector_code	Sector Code	varchar	25			
Mikro_stocks	Mikro_sto_AccGroup_code	Acc. Group Code	varchar	25			
Mikro_stocks	Mikro_sto_packaging_code	Packaging Code	varchar	25			
Mikro_stocks	Mikro_sto_brand_code	Brand Code	varchar	25			
Mikro_stocks	Mikro_sto_size_code	product size Code	varchar	25			

Mikro_stocks	Mikro_sto_color_code	Color Code	varchar	25			
Mikro_stocks	Mikro_sto_model_code	The Model Code	varchar	25			
Mikro_stocks	Mikro_sto_season_code	Season Code	varchar	25			
Mikro_stocks	Mikro_sto_raw_material_code	Raw Material Code	varchar	25			
Mikro_stocks	Mikro_sto_premium_code	Premium Code	varchar	25			
Mikro_stocks	Mikro_sto_quality_control_code	Quality Control Code	varchar	10			
Mikro_stocks	Mikro_sto_package_code	Package Code	Bit				
Mikro_stocks	Mikro_sto_positionFlag_code	Position The Flag Code	Bit				
Mikro_stocks	Mikro_sto_AccCode_artik	product Acc Code Artikeli	Bit				
Mikro_stocks	Mikro_sto_safe_wighed_fl	Goods weighed in the safe?	Bit				
Mikro_stocks	Mikro_sto_size_followUp	Size detailed?	Bit				
Mikro_stocks	Mikro_sto_colordetail	Color Detailed?	Bit				
Mikro_stocks	Mikro_sto_quantity_decimal	Does it produce decimal?	varchar	25			
Mikro_stocks	Mikro_sto_passive	Active/Passive	Float				
Mikro_stocks	Mikro_sto_decreasing_stock	product May Fall To Negative?	varchar	25			
Mikro_stocks	Mikro_sto_custom_tariff_statistical_position	Customs identification Statistics Position Number	Float				
Mikro_stocks	Mikro_sto_point		Tinyint				
Mikro_stocks	Mikro_sto_commission_service_code	The Commission Service Code	Float				
Mikro_stocks	Mikro_sto_commission_rate	Commission Rate	Tinyint				
Mikro_stocks	Mikro_sto_otv_application	ÖTV Application	Tinyint				

Mikro_stocks	Mikro_sto_otv_amount	ÖTV Amount	Float				
Mikro_stocks	Mikro_sto_otv_list	ÖTV Type	Smallint				
Mikro_stocks	Mikro_sto_otv_unit	product ÖTV Unit	Tinyint				
Mikro_stocks	Mikro_sto_premium_rate	Premium-Rate	Float				
Mikro_stocks	Mikro_sto_warranty_period	The Predicted Warranty Period	Float				
Mikro_stocks	Mikro_sto_warranty_period_type	Type Of Warranty Period	Float				
Mikro_stocks	Mikro_sto_fiber_Ne_no	fiber Number	Tinyint				
Mikro_stocks	Mikro_sto_standard_cost	Standard Cost	Bit				
Mikro_stocks	Mikro_sto_Picking_Cash_Amount	product Picking Cash Amount	Float				
Mikro_stocks	Mikro_sto_communication_tax_application	Is there special communication tax application?	Bit				
Mikro_stocks	Mikro_sto_z_report_stocks	Z Report?	varchar	25			
Mikro_stocks	Mikro_sto_max_discount_rate	The Maximum Discount Rate	Tinyint				
Mikro_stocks	Mikro_sto_detail_tracking_InWarehouse control	Detail The tracking Of Warehouse Control?	varchar	25			
Mikro_stocks	Mikro_sto_complementary_code	Complementary Code	Float				
Mikro_stocks	Mikro_sto_auto_barcode_login_form	Automatic Barcode Login Form	Float				
Mikro_stocks	Mikro_sto_auto_barcode_code_strucutre	Automatic Barcode Code Structure	Float				
Mikro_stocks	Mikro_sto_Case_discount_rate	Case Discount Rate	Float				
Mikro_stocks	Mikro_sto_cash_discount_amount	Cash discount Amount	Tinyint				

Mikro_stocks	Mikro_sto_revenue_share	Revenue Share	varchar	25			
Mikro_stocks	Mikro_sto_summary_communication_tax_ammount	Summary Communication Tax Amount	Tinyint				
Mikro_stocks	Mikro_sto_summary_communication_tax_type	Summary Communication Tax Type	Tinyint				
Mikro_stocks	Mikro_sto_expense_code	Expense code	Bit				
Mikro_stocks	Mikro_sto_SCT	(Summary Communication Tax) SCT	Smallint				
Mikro_stocks	Mikro_sto_Deductions_type	Deductions Type	Bit				
Mikro_stocks	Mikro_sto_ExpirDat_fl	Is there an expiry date?	varchar	40			
Mikro_stocks	Mikro_sto_balance_ExpirDat	Balance Expiration Date	varchar	40			
Mikro_stocks	Mikro_sto_Installable_checkout	Installable at checkout?	varchar	40			
Mikro_stocks	Mikro_sto_IFRSdifference_code	difference of (international financial reporting standards) IFRS Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_refund_IFRSdifference_code	Refund IFRS difference Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_domestic_sales_IFRSdifference_code	Domestic Sales IFRS difference Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_sales_refund_IFRSdifference_code	purchasing Refund IFRS difference Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_sales_discount_IFRS difference_code	Sales Discount IFRS difference Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_purchase_discount_IFRSdifference_code	Purchase Discount IFRS difference Acc. Code	varchar	40			

Mikro_stocks	Mikro_sto_sales_cost_IFRSdifference_code	Sales Cost IFRS difference Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_overseas_sales_IFRSdifference_code	Overseas sales IFRS difference Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_additional_expenses_IFRSdifference_code	Additional expenses IFRS difference Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_investment_incentive_IFRSdifference_code	Investment incentive IFRS difference Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_warehouses_inter_sales_IFRSdifference_code	Warehouses inter-sales IFRS difference Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_sales_cost_warehouses_IFRSdifference_code	Cost of Sale Between Warehouses IFRS difference Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_partnership_sales_IFRSdifference_code	Sales of Partnership IFRS difference Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_sales_return_IFRSdifference_code	Sales return to affiliated companies IFRS difference Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_discounted_sales_IFRSdifference_code	Discounted Sale to Affiliated Partnerships IFRS difference Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_diff_sales_price_IFRSdifference_code	Sales Price Differential IFRS difference Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_overseas_sales_cost_IFRSdifference_code	Overseas Sales Cost IFRS difference Acc. Code	varchar	40			
Mikro_stocks	Mikro_sto_partnership_cost_sales_IFRSdiff	Partnerships Based On Cost Of Sales IFRS difference Acc. Code	varchar	40			

	erence_code						
Mikro_stocks	Mikro_sto_zero_cost_sales_IFRSdifference_kod	Zero Paid Cost Of Sales IFRS difference Acc. Code	Float				
Mikro_stocks	Mikro_sto_costt_prod_IFRSdifference_code	Cost Of Production Of IFRS difference Acc. Code	Bit				
Mikro_stocks	Mikro_sto_prod_capacity_IFRS difference_code	Production Capacity Of IFRS difference Acc. Code	Smallint				
Mikro_stocks	Mikro_sto_impairment_IFRSdifference_code	Impairment of IFRS difference Acc. Code	Smallint				
Mikro_stocks	sto_percentage_of_content	Percentage of content	Smallint				
Mikro_stocks	Mikro_sto_will_sent_to_web	Will it be sent to the web	Bit				
Mikro_stocks	Mikro_sto_min_stock_daily_info	Daily information for minimum leveling operation	Tinyint				
Mikro_stocks	Mikro_sto_order_stock_daily_info	Daily information for order leveling operation	Tinyint				
Mikro_stocks	Mikro_sto_max_stock_daily_info	Daily information for maximum leveling operation	Smallint				
Mikro_stocks	Mikro_sto_leveling_opr_evaluation	Will the assessment of the leveling operation be carried out?	bit				
Mikro_stocks	Mikro_sto_otv_deduction_type	Otv deduction type	Tinyint				
Mikro_stocks	Mikro_sto_reso_plan_evaluation	Will be evaluated in the resource planning operation?	tinyint				

Appendix B: System Final Schema Table

Field name	Description_1	Description_2	Syn
Acc_code	Product acc account code		Acc_code, acc_id
Acc_code_discard	Product acc code discarded		Acc_code_discard
Acc_discount_code	Product acc. Discount code		Acc_discount_code
Acc_extra_charges_code	Product acc. Extra charges code		Acc_extra_charges_code
Acc_group_code	Acc. Group code	See. Table stock_acc_groups	Acc_group_code
Acc_overseas_sales_code	Product acc. Overseas sales code		Acc_overseas_sales_code
Acc_purchase_acc_code	Stok acc. Purchasing code		Acc_purchase_acc_code
Acc_purchasing_discount_code	Product acc. Purchasing discount code		Acc_purchasing_discount_code
Acc_return_code	Acc return code		Acc_return_code
Acc_sales_costs_code	Product acc. Sales costs code		Acc_sales_costs_code
Accepted_day1	Goods acceptance day	Monday	Accepted_day1
Accepted_day2	Goods acceptance day	Tuesday	Accepted_day2
Accepted_day3	Good acceptance day	Wednesday	Accepted_day3
Accepted_day4	Good acceptance day	Thursday	Accepted_day4
Accepted_day5	Good acceptance day	Friday	Accepted_day5
Accepted_day6	Good acceptance day	Saturday	Accepted_day6
Accepted_day7	Good acceptance day	Sunday	Accepted_day7
Active	Registration status		Active
Affiliate_sales_cost_acc_code	Affiliate sales cost acc. Code		Affiliate_sales_cost_acc_code
Approved	Approved		Approved, accepted
Attribute	Name and value of the product properties are entered in the field		Attribute
Authority_code	Authorization code		Authority_code
Auto_bar_code_code_structure	Automatic barcode code structure		Auto_bar_code_code_structure
Auto_bar_code_login_form	Automatic barcode login form	0: automatic barcode creation 1: automatic barcode to be created according to the detail tracking 2: create barcode for each entry record	Auto_bar_code_login_form
Automatically_increase_product_serial_number	Automatically increase product serial number		Automatically_increase_product_serial_number
Balance_expiry_date	Balance expiration date		Balance_expiry_date
Brand_code	Brand code	See. Table stock_brands	Brand_code
Bundle	Bundle products		Bundle, package
Can_use_interns	Used in movements (1: true 2: false)	Can_use_interns	
Cancel			Cancel, drop
Case_discount_rate	Case discount rate		Case_discount_rate
Cash_discount_amount	Cash discount amount		Cash_discount_amount
Category_code	Product category code		Category_code
Changed			Changed, improved, altered
Check_sum			Check_sum

Class_code	Class code		Class_code
Class_type	Material class type		Class_type
Colour_code	Color code	See. Table stock_color_definitions	Colour_code
Colour_detail	Color detailed?	0: yes 1: no	Colour_detail
Comb_lot_units	Lot sizes combineable		Comb_lot_units
Commission_rate	Commission rate		Commission_rate
Commission_service_code	The commission service code		Commission_service_code
Communication_tax_application	Is there special communication tax application?	0: no 1: yes	Communication_tax_application
Complementary_code	Complementary code		Complementary_code
Condition	Product state	Condition, state	
Cost_export_sales_acc_code	The cost of export sales acc. Code		Cost_export_sales_acc_code
Create_user			Create_user
Created_by	Created by		Created_by
Created_date	Created date		Created_date
Created_hour	Created hour		Created_hour
Created_min	Created minute		Created_min
Created_sec	Created second	Created_sec	
Currency_type	Product list price currency	Currency_type	
Custom_tax_statistical_position	Customs identification statistics position number		Custom_tax_statistical_position
Data_ref	Data reference		Data_ref
Date_of_production	Product production date (dd / mm / yyyy)		Date_of_production
Date_of_update			Date_of_update
Decreasing_stock	Product may fall to negative?		Decreasing_stock
Deductions_type	Deductions type	0: withholding 1: withholding 31 2: withholding 91 3: withholding 21 4: withholding 32 5: withholding 61 6: withholding 45 7: withholding full	Deductions_type
Depr_dur2	Depreciation duration2		Depr_dur2
Depr_rate	Depreciationrate		Depr_rate
Depr_rate2	Depreciation rate2		Depr_rate2
Depr_type	Depreciationtype		Depr_type
Depr_type2	Depreciation type2		Depr_type2
Depreciation_duration	Depreciation duration		Depreciation_duration
Description	Product description information (can be html)		Description
Detail_tracking_in_warehouse_control	Detail the tracking of warehouse control?		Detail_tracking_in_warehouse_control
Diff_sales_price_acc_code	The difference in the sale price acc. Code	Diff_sales_price_acc_code	
Discount	Product discount information		Discount
Discount_can't_done	Discounts can't be done?	0: yes 1: no	Discount_can't_done
Discount_price			Discount_price
Dist_amount	Distributed amount		Dist_amount

Dist_lot_units	Lot size can be distributed	Dist_lot_units	
Divided_lot_size	Lot size can be split		Divided_lot_size
Dominant_code	Materialcard code		Dominant_code, dominant_id
Electronic_label_type	Electronic label type	0: standard label 1: small sticker 2: fruit and vegetable label	Electronic_label_type
Expense_code	Expense code		Expense_code
Expiration_date	Product expiration date (dd / mm / yyyy)		Expiration_date
Expiry_date	Is there an expiry date?	Expiry_date, end_date	
Fibre_no	Fiber number	Fibre_no, fibre_code	
File_id			File_id, file_code
Fixed_lot_size	Fixed lot size		Fixed_lot_size
Following_details	Following details	0: no detail tracking 1: party basis 2: party lot basis 3: serial number basis 4: 5 on the basis of bond	Following_details
Foreign_name			Foreign_name, foreign_label, foreign_tag
Given_order_unit	Given order unit		Given_order_unit
Group_id			Group_id
Hidden			Hidden, concealed
Id	Product category number		Id, code
In_liquidation	Short-lived provisional all?	0: yes 1: no	In_liquidation
Installable_at_checkout	Installable at checkout	Installable_at_checkout	
Inventory_subgroup_code	Inventory subgroup code	See. Table stock sub groups	Inventory_subgroup_code
Inventory_tracking	Product inventory tracking		Inventory_tracking
Investment_promo_acc_code	Investment promotion of acc. Code	Investment_promo_acc_code	
Label_account	Print a label?	0:did not print 1:print	Label_account
Last_up_date			Last_up_date
Last_up_user			Last_up_user
Levelling_operation_evaluation	Will the assessment of the leveling operation be carried out?		Levelling_operation_evaluation
Locked			Locked
Lot_sizing_mtd	Lot determination method		Lot_sizing_mtd
Max_discount_rate	The maximum discount rate		Max_discount_rate
Max_order_qty	Max order quantity		Max_order_qty
Max_stock			Max_stock
Max_stock_daily_info	Daily information for maximum leveling operation		Max_stock_daily_info
Max_stock_level	Maximum product level		Max_stock_level
Min_order_qty	Minimum order quantity		Min_order_qty
Min_stock	The minimum product level		Min_stock
Min_stock_daily_info	Daily information for minimum leveling operation		Min_stock_daily_info

Model_code	The model code	See. Table stock_model_definitions	Model_code
Modified_by	Modifed		Modified_by
Modified_date	Modifed date	Modified_date	
Modified_hour	Modified hour	Modified_hour	
Modified_min	Modifedminute		Modified_min
Modified_sec	Modifed seconds		Modified_sec
Mult_order_qty	Multi order quantity		Mult_order_qty
Name			Name, label, tag
O_t_v_amount	Ötv amount		O_t_v_amount
O_t_v_application	Ötv application	0:ötv no 1:receipt from the amount 2:meet the percentage 3:from the amount on sale 4:sales percentage 5:receipt and sales amount 6:receipt and sales percentage	O_t_v_application
O_t_v_deduction_type	Otv deduction type	0:no deduction 1:withholding	O_t_v_deduction_type
O_t_v_list	Ötv type	0:no 1:ötv1 2:ötv2 3:ötv3 4:ötv4 5:ötv3a 6:ötv3b 7:ötv3c	O_t_v_list
O_t_v_unit	Product ötv unit		O_t_v_unit
Option_price	List price of the product product unit		Option_price
Order	Product image display order		Order
Order_stock_daily_info	Daily information for order leveling operation		Order_stock_daily_info
Order_time	Order time (days)		Order_time
Orders_day1	Order days	Monday	Orders_day1
Orders_day2	Order days	Tuesday	Orders_day2
Orders_day3	Order days	Wednesday	Orders_day3
Orders_day4	Order days	Thursday	Orders_day4
Orders_day5	Order days	Friday	Orders_day5
Orders_day6	Order days	Saturday	Orders_day6
Orders_day7	Order days	Sunday	Orders_day7
Package_code	Package code	See. Table stock_package_definitions	Package_code
Packaging_code	Packaging code	See. Table stock_packagins	Packaging_code
Parent_group_code	Product of the parent group code	See. Table stock main groups	Parent_group_code
Parent_group_code	Product of the parent group code	See. Table stock main groups	Parent_group_code
Part_dep	Part depreciation		Part_dep
Partdep2	Part depreciation2		Partdep2
Passive	Active/passive	0: passive 1: active	Passive
Percentage_of_content	Percentage of content	Percentage_of_content	
Photo_name			Photo_name
Picking_cash_amount	Product picking cash amount		Picking_cash_amount
Point			Point
Position_flag_code	Position the flag code		Position_flag_code
Premium_code	Premium code		Premium_code
Premium_rate	Premium-rate		Premium_rate
Preparing_day	Production time to ship (in days)		Preparing_day

Price	Product base price	Price	
Producer_code	Manufacturer code		Producer_code
Product_code			Product_code, stock_code, item_code
Product_location_store_management	Place of use - store management		Product_location_store_management
Product_officer_code	Product officer code	See. Table staff	Product_officer_code
Product_place_of_purchase	Place of use - purchasing		Product_place_of_purchase
Product_place_of_sales_and_distribution	Place of use - sales and distribution		Product_place_of_sales_and_distribution
Product_type	0: commercial goods 1: first article 2: intermediate product 3: semi-finished product 4: product 5: side product 6: operating material 7: consumption material 8: spare part 9: fuel stock 10: installation prescription product 11: basic raw material		Product_type
Production_global_commercial_item_number	Production global commercial item number		Production_global_commercial_item_number
Production_global_trading_item_number	Production global trading item number		Production_global_trading_item_number
Production_manufacturer_part_number	Production manufacturer part number		Production_manufacturer_part_number
Profit	Profit rate	Profit	
Quality_control_code	Quality control code	See. Table stock_quality_control_definitions	Quality_control_code
Quantity	The amount of product	Quantity, amount	
Raw_material_code	Raw material code	See. Table stock_main_raw_material	Raw_material_code
Rec_id_rec_no			Rec_id_rec_no
Rec_no			Rec_no
Received_order_unit	Received order unit		Received_order_unit
Resource_plan_evaluation	Will be evaluated in the resource planning operation?	0:true1:false	Resource_plan_evaluation
Retail_rate	Retail vat rate		Retail_rate
Rev_depr_flag	Revaluation depreciation	Rev_depr_flag	
Rev_depr_flag2	Alternative valuation depreciation		Rev_depr_flag2
Revaluation_flag	Revaluation		Revaluation_flag
Revaluation_flag2	Revaluation 2		Revaluation_flag2
Revenue_share	Revenue share		Revenue_share
Safe_weighed	Goods weighed in the safe?	0: yes 1: no	Safe_weighed
Sales_cost_stores_acc_code	Cost of sales between stores of acc. Code		Sales_cost_stores_acc_code
Sales_end_date	Product sales end date (dd / mm / yyyy)		Sales_end_date
Sales_price			Sales_price
Sales_start_date	Product sales start date (dd / mm / yyyy)		Sales_start_date
Sales_stop	Sales stop?	0:did not stop 1:stop	Sales_stop
Salvage_value	Salvage value		Salvage_value

Season_code	Season code	See. Table stock_year_season _definitions	Season_code
Sector_code	Sector code	See. Table stock_sectors	Sector_code
Seller_code			Seller_code
Seller_store_code	Store product code		Seller_store_code
Shelf_label	Shelf label	0:no 1:yes	Shelf_label
Shelf_life	Shelf life		Shelf_life
Shipment_template	Delivery template name		Shipment_template
Short_name	Short name	Short_name	
Site_id	Data center	Site_id	
Size_code	Product size code	See. Table stock_size_definiti ons	Size_code
Size_followup	Size detailed?	0: yes 1: no	Size_followup
Spec_code	Special code		Spec_code
Spec_rec_no			Spec_rec_no
Special1			Special1
Special2			Special2
Standard_cost	Standard cost		Standard_cost
State			State
Stock_name	Product name	Stock_name, product_name, item_name	
Stock_order	Order time (days)		Stock_order
Stock_pieces			Stock_pieces
Stock_retail_tax	Retail tax rate	Stock_retail_tax	
Stop_accepted_goods	Will you accept the goods?	0:did not stop 1:stop	Stop_accepted_goods
Stop_order	Stop order?	0:did not stop 1:stop	Stop_order
Store_product_code	Store product code		Store_product_code
Stores_sales_acc_code	Stores sales acc. Code		Stores_sales_acc_code
Sub_group_no	Sub group number	Sub_group_no	
Subtitle			Subtitle
Summary_communicati on_tax	(summary communication tax) sct		Summary_communicatio n_tax
Summary_communicati on_tax_amount	Summary communication tax amount		Summary_communicatio n_tax_amount
Summary_communicati on_tax_type	Summary communication tax type	0:none 1:sct 2:5035 nambered by the low of sct	Summary_communicatio n_tax_type
Tax			Tax
Title			Title
Tool	Tool		Tool
Track_type	Track type		Track_type
Uni_vid	Out of use		Uni_vid
Uniform_resource_locat er	Product official url	Uniform_resource_locat er	
Unit1_coefficient	Unit1 coefficient		Unit1_coefficient
Unit1_height	Unit height (mm)		Unit1_height
Unit1_length	Unit length (mm)		Unit1_length
Unit1_name	Unit name		Unit1_name
Unit1_tare	Unit1 tare		Unit1_tare
Unit1_weight	Unit net weight (kg)		Unit1_weight
Unit1_width	Unit width (mm)		Unit1_width
Unit2_coefficient	Unit coefficient	Unit2_coefficient	
Unit2_height	Unit height (mm)		Unit2_height
Unit2_length	Unit length (mm)		Unit2_length
Unit2_name	Unit name		Unit2_name
Unit2_tare			Unit2_tare

Unit2_weight	Unit net weight (kg)		Unit2_weight
Unit2_width	Unit width (mm)	Unit2_width	
Unit3_coefficient			Unit3_coefficient
Unit3_height	Unit height (mm)		Unit3_height
Unit3_length	Unit length (mm)		Unit3_length
Unit3_name	Unit name		Unit3_name
Unit3_tare			Unit3_tare
Unit3_weight	Unit net weight (kg)		Unit3_weight
Unit3_width	Unit width (mm)		Unit3_width
Unit4_coefficient	Unit coefficient		Unit4_coefficient
Unit4_height	Unit height (mm)		Unit4_height
Unit4_length	Unit length (mm)		Unit4_length
Unit4_name	Unit name		Unit4_name
Unit4_tara			Unit4_tara
Unit4_weight	Unit net weight (kg)		Unit4_weight
Unit4_width	Unit width (mm)		Unit4_width
Warehouse_code	Warehouse address		Warehouse_code
Warranty_period	The predicted warranty period		Warranty_period
Warranty_period_type	Type of warranty period	0: month 1: day 2: year	Warranty_period_type
Wholesale_rate	Wholesale vat rate		Wholesale_rate
Will_sent_to_web	Will it be sent to the web		Will_sent_to_web
Yield	Yield		Yield, output
Z_report	Z report?		Z_report
Zero_paid_cost_sales_acc_code	Zero paid cost of sales acc. Code		Zero_paid_cost_sales_acc_code