International Journal on Advanced Science Engineering Information Technology

Vol.9 (2019) No. 1
ISSN: 2088-5334

Lexical Disambiguation (CKBD): A Tool to Identify and Resolve Semantic Conflicts Using Context Knowledge

Said Al Tahat[#], Kamsuriah Ahmad[#]

[#] *Center for Software Technology and Management, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600, Bangi, Selangor, Malaysia*
E-mail: saed_tahat@yahoo.com, kamsuriah@ukm.edu.my

***Abstract*— The schema matching process is a fundamental step in a schema integration system, and its quality impacts the overall performance of the system. Recently, a large number of schema matching approaches have been developed. Until today, the performance of schema matching is inherently uncertain and requires improvement. The most difficult task is inferring the real-world semantics of data from the information provided by schema labels in their representations. Usually, schemas with identical semantics are represented by different vocabularies and only their own designers can completely understand. A schema may contain synonyms and homonyms words. Therefore, it is necessary to understand how the schema elements are "presented"; it is often hard to get aware meaning associated with elements names, due to the semantic ambiguity of human language. Semantic ambiguity problem means the capability of being understood in two or more possible senses. Having more than one meaning for an individual schema element would cause confusion in interpretation of schema name. This may affect negatively on the matching result. Therefore, this paper aims to resolve this problem of semantic ambiguity and represent the intended meaning of the schema labels name, by introducing the CKBD (Context Knowledge-Based Disambiguation) approach. The CKBD is obtained by integrating two pieces of context knowledge: semantic domain and more frequency used into a disambiguation processor. Finally, the CKBD is implemented and is tested in a real dataset. The result is deeply grounded in the ability to detect schema name intended meaning.**

***Keywords*—natural language processing; semantic ambiguity; schema integration; schema matching; word sense disambiguation.**

I. INTRODUCTION

The integration of data sources that are autonomously established needs to take into account the issue of heterogeneity[1]. Fortunately, there are now readily available standard solutions to deal with the conflicts that emerge from the technical and data model heterogeneity. Somehow, despite years of research, semantic heterogeneity resolution is still an open issue. Schema integration (SI) is a system that addresses this problem [1]-[3].

Schema integration (SI) is a technology that addresses the problem of schema heterogeneity by creating a correct, complete, minimal, and understandable unified global schema of the existing or proposed databases [4]. The SI system receives some locale source schemas as input and then creates an integrated schema as output, which is called a global schema, with the mappings ruled to the local databases. In recent years, SI has appeared to be an efficient tool that has enabled the sharing of information among heterogeneous and autonomous databases. It has also provided transparent access to remote data [5].

The major aim of schema integration is to provide users with a unified global schema, where users can access, retrieve, and utilize the information, instead of relying on a list of database schema [6]. Natural Language Processing (NLP), schema matching, Information Retrieval (IR), Information Extraction (IE), and named-entity recognition are combined in the development of SI technology [7]. The architecture of an SI system consists of four major modules: pre-integration processing, schema matching processing, conflict solving processing, and merging processing. Most SI systems will follow to some extent the same pipeline structure illustrated in Figure 1.

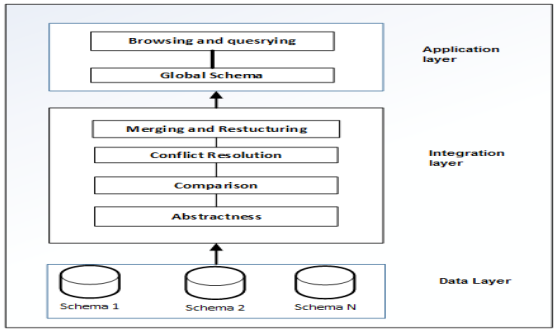

Fig. 1: Schema Integration Architecture [8]

The most challenging problem in developing a SI system is schema matching, which is the process of discovering the right semantic that corresponds with schemata from different sources during the generation of global schema [9], and this is the major reason why the currently employed methods still encounter several obstacles, such as semantic ambiguity [1, 10, 11].

The process of finding similarities between the schema elements of heterogeneous data sources is known as schema matching [12]. Recently, a large number of schema matching approaches have been developed [13], achieving impressive performance on some datasets, but in general, this method is not expected to yield correct results. Even now, the performance of schema matching is inherently uncertain and requires improvement [13]. Moreover, several problems remain or have only been partially solved. Furthermore, even though currently the schema matching process has improved, it is not entirely automated, has inadequacies in numerous areas, and needs improvements that must consider an increasing amount of data, schema, and data sources[13].

Differences in the meaning of schema elements (or semantic heterogeneity of data sources) are the main problems handled by automatic or semi-automatic schema matching [1]. Usually, data sources are developed by people, based on different organizational demands, and differing mostly in descriptions and expressiveness that lead to increasing semantic heterogeneity [14]. Furthermore, schemas with identical semantics are represented by different vocabularies that only their designers can completely understand, where some real-world entities are represented in both databases by ambiguous words [15]. For example, a schema may be represented by a synonym word (when different words are used to represent the same element) such as the element *Writer* and *author*, or by homonym words (when the same words are used to name different elements). Synonyms and homonyms can mislead the process of schema matching [16-19].

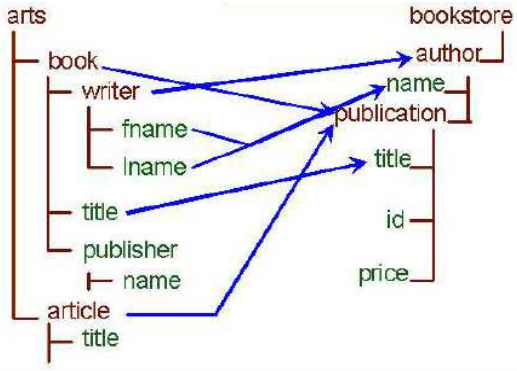

Fig. 2 Dataset for Book

Synonyms and homonyms are among the lexical semantic ambiguities in a language's inherent concepts (names), which are utilized in a database schema and the associated meanings represented in an organization [1, 19, 20]. Ambiguity happens when a word can be understood in two or more possible senses. This is a pervasive phenomenon among these database schemas that prevents accurate matching, due to the increment of the number of wrongly matched candidates [1]. As mentioned in one study [21], finding a word with only one meaning is not easy because a word may have more than one interpretation. For example, the word "Area" has several distinct lexical definitions, including "a subject of study" or "a particular environment or walk of life," as shown in Figure 3.

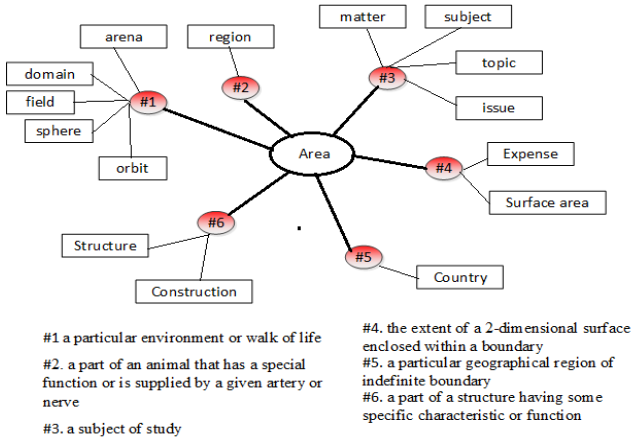

Fig. 3 The word "area" with a related sense

Given the context in which an ambiguous word is used, anyone can easily recognize its meaning [22]. However, for computers, this task is intensely difficult. This is because of erroneous results and ridiculous conclusions, which will negatively affect the matching result, could result from an individual word with more than one meaning, as it is often difficult for the computer to be aware of the intended meaning of schemata elements. Therefore, it is necessary to deal with the semantic ambiguity problem of how the schema elements are "represented," which makes lexical disambiguation crucial to understand [23].

Therefore, the meanings of schema labels must be determined in the schema matching process [1]. The correct assignation of meaning to every schema label— concerning the semantic resource (a well-known semantic resource being WordNet)—enables the discovery of official semantic correspondences among the elements of different schemas [10]. This identification requires applying the word sense disambiguation (WSD) technique. The quality of semantic correspondence accuracy in schema matching is expected to improve with the use of WSD. However, because WSD is still an unsolved problem in NLP (natural language processes) [24], its direct application to data sources that are structured and semi-structured is out of the question [10].

Thus, this paper aims to resolve the lexical semantic ambiguity problem in structured and semi-structured data sources by devising the CKBD (context knowledge-based disambiguation) approach. This approach reuses some well-established ideas from the dictionary-based methods and is considered a knowledge-based method because it exploits the semantic description of senses in the lexical source. CKBD is obtained by integrating two pieces of context knowledge: semantic domain and more frequency used in a disambiguation processor. Context knowledge resource is modeled to disambiguate lexical words. First, the context selection processor determines the disambiguation context based on neighbourhood words in a schema. Second, the intended meaning can be determined from a set of meanings

using contextual information. Context knowledge consists of words with their senses obtained from WorldNet. Consequently, in order to evaluate the proposed processor, it is first applied to a schema integration system. More details will be presented in the problem statement section.

II. Material and Method

In a typical dictionary, a word may have several different meanings. For instance, a bank can be defined as "the edge of a river," or "a financial institution" while a pen could mean "a livestock's enclosure" or "a writing instrument." In linguistics, this phenomenon is termed as lexical semantic ambiguity. As per Weaver (1955), initially, the issue of lexical ambiguity grabbed the attention of scholars in the domain of machine translation. This is because, in the output language, the different senses of an ambiguous word may require interpretations into differing meanings. In the context of computer systems, this necessitates the search for the correct sense where the context applies a word. About this, Ponzetto and Navigli (2010) gave rise to the term word sense disambiguation (WSD) as the process of defining the sense of a word in the context of a particular natural language.

A. Word sense disambiguation

WSD encompasses a classification task in which word senses become the classes. Furthermore, a technique of automatic classification is employed for assigning each word manifestation to one or more classes according to the evidence [24]. The evidence is derived from the context and knowledge from outside sources. Thus, WSD is about determining the connection existing amongst "word" and "meaning" and "context."

Context is the only way of identifying the meaning of an ambiguous word. As such, having contextual information to enable the determination of the intended meaning from a set of meanings is critical. According to [23], the two ways of determining context are relational information and bag of words. Relational information denotes ambiguous word relations, and this information comprises syntactic relations, semantic categories, selection preferences, phrasal collocation, orthographic properties, and distance from the target. Meanwhile, a bag of words denotes words in a given neighbourhood with no consideration of their associations with the ambiguous word.

Numerous researchers including [20]-[22], have proposed several WSD approaches. The algorithms of WSD are grouped into three key WSD approaches as explained below. These include 1) knowledge-based or dictionary-based methods and 2) corpus-based approaches.

Knowledge-based or dictionary-based methods require the use of thesauri, electronic dictionaries, and lexical knowledge bases, but corpus evidence is not needed. Usually, these methods depend on the computation of similarity measures. According to the available literature, two methods exist in the knowledge-based approach, which is AI-based methods, and dictionary-based methods.

Corpus-based approaches require the use of machine learning methods. The corpus-based method entails a disambiguating strategy by the information that is directly extracted from textual data. This method requires the use of obtained information through the training of the models of statistical language on a corpus. The algorithms employed in the corpus-based approach fall into either: 1) supervised algorithms; or 2) unsupervised algorithms.

Supervised algorithms are usually applied when there is a set of manually hand-labeled instances, also known as a training set. Therefore, by employing the training sets, these instances are trained before being used for classifying a set of unlabelled examples known as the test set. Meanwhile, the unsupervised algorithms manipulate corpora that are unlabelled, and the raw corpora or knowledge base. These algorithms do not necessitate hand-labeled corpus in offering a sensible choice for a word in context. Instead, Ponzetto and Navigli [25] stated that these methods are grounded on the notion that a word of a similar sense will have similar neighboring words.

B. WSD in structured and semi-structured data sources

Traditionally, WSD is applied to plain text. Here, Rachman and Saptawi [16] stated two methods for determining the context. The first method employs relational information, which is about the relationships of ambiguous words. These include semantic categories, parts of speech, discourse, phrasal collocation, and syntactic features. The second method employs a bag of words, which concerns words in the neighbourhoods in the text. Here, their relationships with the ambiguous word are not taken into account [1]. However, in some structured and semi-structured data sources, such features are unavailable.

Furthermore, the context comprises words in schemata functioning as classes and name of attributes or relationships among schema. Moreover, most words in a schema are a member of the syntactic noun category. This is beyond the use of semantic annotations [10, 23].

Schema name definition is implicit and loose, and therefore, there may still be ambiguity [10], [21]. Thus, through the schema name, disambiguation will obtain context that is unstructured and highly heterogeneous. This is usually in the form of free text in which the application of syntactic analysis is impossible due to the existence of poorly-formed sentences. This often refers to designers' subjective impressions such as the word "client" or "costumer" or technical details such as "Nikon" or "photo." As such, the problem of determining the context words that better help the disambiguation arises, as many user tags are useless (or even harmful) for disambiguation.

C. WSD Evaluation Measures

The existence of diverse resources of knowledge adopted, test sets, and sense inventories, make the comparison and evaluation of WSD systems very challenging. As an example, WSD systems involve the use of diverse text types such as domain-specific texts or highly technical texts where the employed senses are restricted. On the other hand, the employed senses may be more flexible in general texts.

In itself, WSD is not an end to a process. Instead, WSD is a common task that is crucial to the entire task including machine translation and information retrieval. Therefore, there are two probable types of WSD work evaluation: 1) *in vitro* evaluation; and 3) *in vivo* evaluation. In *in vitro* evaluation, the WSD systems are independently tested using benchmarks that are specially constructed. On the other hand,

in *vivo* evaluation, instead of being independently tested, outcomes are evaluated based on their contribution to the system's entire performance for a specified application (e.g., schema matching systems).

This paper follows the extrinsic (*in vivo*) evaluation. It attempts to solve lexical ambiguity when no other features other than an unstructured set of words are available. Accordingly, an intelligent disambiguation approach is introduced to solve this problem. The proposed approach takes into account the context selection problem as a critical aspect of WSD when applied in structured and unstructured sources.

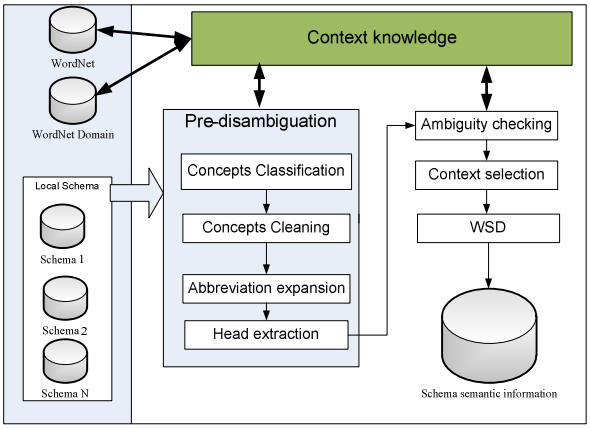

Fig. 4 CKBD Context knowledge-based disambiguation Approach

The proposed approach comprises four major phases, which are: (1) pre-disambiguation, (2) ambiguity checking, (3) context selection, and (4) disambiguation. Figure 4 illustrates the architecture of the proposed approach. A more detailed process of these phases is explained in the following sub-sections.

A. Phase 1: Pre-disambiguation

The database designers typically use different terminology to represent the schema name. In other words, a different representation of "names" of columns or tables may contain a schema. Names consisting of more than one token (word) with a different order of tokens, e.g. birthdate vs. date birth are an example of a schema. This includes names with stop words such as name_of_department vs. Department Name and names that are abbreviated vs. extended such as PO vs. Purchase Order. Therefore, the pre-disambiguation process is primarily performed to reduce the noise in the name obtained from an unusual Word. In this step, some preparation steps are performed in the listed sequences, which are: Concept classification, Abbreviation expansion, Concept cleaning, and Head extraction.

B. Phase 2: checking ambiguity

In this approach, the context knowledge is accessed to provide a set of candidate senses related to the word. Context knowledge is discussed in detail in Part E. After that, ambiguity checking will determine if the input word is ambiguous or not, by checking the number of related senses. If the word has a single word-sense, then the word is unambiguous, and vice versa.

C. Phase 3: Context Selection

Given a target keyword to disambiguate TW, disambiguation context is defined based on tags. For example, if the structure relationship tags of the target word are a TN, then, all words in the schema tagged by TN are selected as the disambiguation context. However, the disambiguation context is not constrained to contain all the words tagged with TN in the schema. In this paper, the proposed method follows the contemporary methods' context size, with three surrounding words for the window of context.

D. Phase 4: Disambiguation

The way in which context knowledge can be used for WSD is described in this section. A context knowledge-based approach is proposed because the meaning of an ambiguous word can only be identified from its context. The input to the ambiguous word W, its set of possible senses SW, and disambiguation context CW in the bag-of-words approach is the disambiguation phase. The output will be the proper senses of the ambiguous word according to the context. The combination of two different types of context knowledge, domain name, and the frequency of usage knowledge, are used as the basis for the proposed disambiguation algorithm. Combinations of different types of knowledge effectively improve the performance of WSD.

First, all possible senses associated with a term are examined, and the domains connected to these senses are extracted to find the domain for each term in the context. The algorithm does this process. Next, a list of the more various domains of the term is computed. However, the ultimate selection for the number of more many domains has yet to be solved. In this research, the algorithm chooses the first three most frequent domains for the individual term in the context.

Second, the context domain is determined. Here, all possible domains associated with a term in the context is examined by the algorithm. Subsequently, the algorithm computes a list that contains the more common domains in the context. In this step, the more constant domains are chosen as the context domain.

Finally, the context domain list is compared with the domains associated with target word senses. Here, all the senses associated with the context domains are chosen. Finally, the correct sense of the target word is determined, which the sense is belonging to more frequency of use.

This scenario assumes that the target word W and the context words C1, C2, C3, C5 are received as input; then, the domains of target word B1 = (d1, d2, d3, d…n) is contained in bag B1. Meanwhile, the sets of all domains corresponding to the context words, C1, C2, C3, C…n, are contained in B2. All possible domains corresponding to the bag B1 domain is contained in each set Ci.

- Input the ambiguous word, its senses, and the disambiguation context.
- Insert the senses corresponding to all targets with its domain into bag B1.
- Find a list of the more common domains for each word in the context, and choose the first three most frequent domains and insert into bag B2.

- The domain in B2, which maximizes the domains of other content words, is the domain of the context.
- Insert into B3 the sense belonging to the context domain obtained from Step 4.
- The sense in B3 with more frequency of use is the correct sense.
- The output of this phase is a term with a set of candidate senses and its knowledge, which will be saved into an internal dataset (context knowledge).

E. Context Knowledge

In a schema, the context of the word determines its meaning—whether correct or not. Other words in the neighborhood in the relationships that describe inter- and intra-schema (or also called local context or sentential context) are used to determine the context. In disambiguating words using the context knowledge, various kinds of information are used. Therefore, a context knowledge resource is modeled to disambiguate lexical words. Context knowledge contains a set of words labeled with their senses and domain labels and the more frequencies of usage. A set of words, in which each word has a strong semantic relation to the other, is called a domain. Fixing the target word domain is done based on the content word domains in the local context.

A potential domain has been assigned to each sense. In Table 1, for instance, the word 'bank' has ten meanings, and each sense is labeled with a domain that may indicate its potential context. In determining the correct context for the ambiguous word, the proposed processor uses unambiguous neighborhood words in the schema [22]. From Table 5, it can be observed that, for one word, one domain label can be assigned to multiple senses. For example, the domain labelled "Economy" is assigned to the word group consisting of "bank#1", "bank #3", "bank #4", "bank #6", "bank #7", whereas "bank#2" and "bank#7" are grouped into the domains labelled "Geography" and "Geology." Therefore, context knowledge is modeled from knowledge of frequencies of use to determine the correct meaning, as shown in Table 1.

TABLE I
CONTEXT KNOWLEDGE OF THE WORD BANK

	Sense	Domain	Freq. of use
#1.	sloping land	Geography, Geology	25
#2.	financial institute	Economy	20
#3.	a long ridge or pile	Geography, Geology	2
#4.	container	Economy	0
#5.	the funds held by a gambling house	Economy, Play	0
#6.	a flight maneuver	Transport	0
#7.	a supply or stock held in reserve	Economy	0
#8.	a building in which the business of banking is transacted	Architecture, Economy	0
#9.	Bank building	Architecture, Economy	0
#10	Bank, cant, camber (a slope in the turn of a road)	Architecture	0

Figure 5 shows a purchase order schema in which a user wanted to find the intended meaning of order. Based on context knowledge, the order has ambiguous meanings because it has ten senses, where each sense is labeled with a domain, which could be the potential context.

.

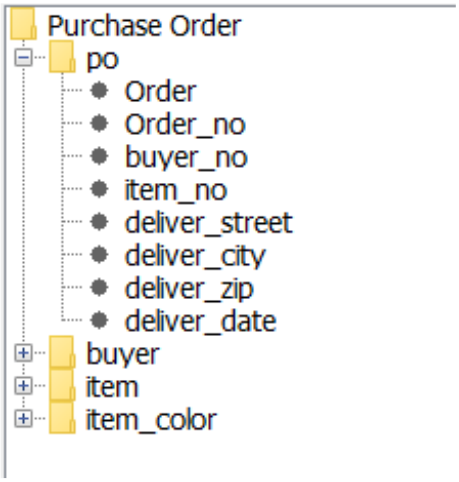

Fig. 5: Purchase Order Schema

The diagram in Figure 5 also shows the context selection phase, where only certain content words—such as purchase order, item, and buyer—are selected as the disambiguation context. Figure 6 shows the disambiguation context words, and the domains and sense numbers of the targeted words.

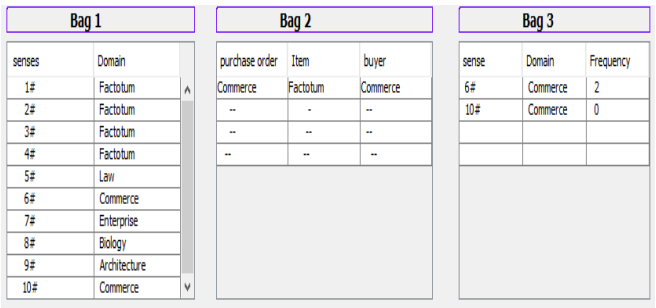

Fig. 6 : Contents of bags

The following are the given examples as shown in Fig. 6.

b1 = the target word sense with its domain
b2 = the disambiguation context with its domain (purchase order, item, and buyer)
b3 = the sense (6# and 10#) belonging to the disambiguation context domain (Commerce)

'Commerce' from bag 2, which achieved a maximum score (factotum), was selected as the domain of the disambiguation context. However, two senses were found to belong to the commerce domain. In this case, the noun sense that belongs to commerce domain, which has a higher frequency of usage, is selected as the correct sense.

\

III. Results and Discussion

The experiments executed in this paper aim to show how the intended meaning of an ambiguous word can be automatically and correctly identified using CKBD. Therefore, we follow the extrinsic (*in vivo*) evaluation. Furthermore, CKBD was evaluated as a stand-alone system using the quality measures proposed by []. Only the accuracy measures are taken into consideration in the CKBD system evaluation. For this purpose, real data schemas from Purchase Order domains are used with two schemas consisting of 60 columns (26 of which were ambiguous) making up each data set.

Gold standards, manually generated by a human expert, were created to assess the quality of the CKBD method. The corresponding gold standard was compared to the results obtained from the experiment. In carrying out the experiments, WordNet 2.0, a lexical database, WordNet Domains 3.2, an extension of the lexical database, and the Abbreviations dictionary, were used as external sources.

Finally, the CKBD method was successfully implemented and tested. A sample of gold standards and the schema that the CKBD approach has handled are presented in Table 2.

TABLE II
A SAMPLE OF GOLD STANDARDS

Headword	**Senses number**	**Correct sense**	**glosses**
Customer	1	#1	someone who pays for goods or services
number	12	#2	a concept of quantity derived from zero and units
city	3	#1	a large and densely populated urban area
zip code	1	#1	a code of letters and digits added to a postal address to aid in the sorting of mail
telephone	2	#1	an instrument that converts sound into signals that can be transmitted over distances and then converts the received signals back into sounds
product	6	#1	commodities offered for sale
price	6	#1	the amount of money needed to purchase something
stock	17	#3	the merchandise that a shop has on hand
Purchase order	1	#1	a commercial document used to request someone to supply something in return for payment
date	8	#2	a particular day specified as the time something will happen
zip	1	#1	a code of letters and digits added to a postal address to aid in the sorting of mail
line	29	#22	a particular kind of product
quantity	3	#2	an adequate or large amount
supplier	1	#1	someone whose business is to supply a particular service or commodity
item	5	#3	an individual unit; especially when included in a list or collection
cost	3	#3	the value measured by what must be given or done or undergone to obtain something
stock	17	#3	the merchandise that a shop has on hand
id	2	#1	a card or badge used to identify the bearer
description	3	#3	Sort or variety; "every description of the book was there."

During the evaluation, an accuracy metric was employed to calculate the results. The accuracy metric employed in the evaluation of sense disambiguation system comprises simple accuracy, which implies word percentage of the accurately disambiguated words. Hence, an accuracy metric is an integral metric for measuring the disambiguation's performance. Accordingly, the manner in which accuracy is computed is expressed in Equation (1) below:

$$Accuracy = \frac{\#\,\text{correctly disambugated words}}{\#\,\text{ambigous words}} \quad (1)$$

If the sense that CKB selects is the same as the one that the gold standard returns, then the disambiguated concepts are considered to be correctly disambiguated and vice versa. Furthermore, a first-sense baseline will be applied to the dataset initially, and then a comparison regarding the accuracy is made, as outlined in Table 3.

TABLE III
EVALUATION OF THE CKBD METHOD

Headword	**Senses number**	**Correct sense**	**CKBD**	**first-sense**
customer	1	#1	✓	✓
number	12	#2	✓	--
city	3	#1	✓	✓
telephone	2	#1	✓	✓
product	6	#1	✓	✓
price	6	#1	✓	✓
stock	17	#3	✓	--
purchase order	1	#1	✓	✓
date	8	#2	✓	--
line	29	#22		
quantity	3	#2		✓
supplier	1	#1	✓	✓
item	5	#3	✓	--
cost	3	#3	--	--
stock	17	#3	--	--
id	2	#1	✓	✓
description	3	#3	✓	--

The test set of Purchase Order database schema contains 60 ambiguous words; the first-sense baseline correctly

disambiguated only 31 words on WSD. Thus, based on Equation (1), the accuracy of disambiguation is 55.0%. In the CKBD approach, only 55 words were truly disambiguated. Thus, based on Equation (1), the accuracy of disambiguation is 83.3%. The reason behind the low accuracy of the first-sense baseline on WSD in the context of this dataset is because the first-sense information is not given for obsolete words; only the first-sense that each word has in modern English (from 1970 up to the present day) is indicated. The calculation, in this case, is performed by taking the frequencies for each decade (from 1970 up to the present day) and averaging them. Table 4 shows the disambiguation result of the test set of two database schemas obtained by using the first-sense baseline on WSD and CKBD.

TABLE IV
EXPERIMENTAL RESULT

	CKBD	**first-sense WSD**
Ambiguous words	60	60
Correctly disambiguated words	55	31
Accuracy	83.3%	55.0%

IV. CONCLUSIONS

Finding the meanings of schema labels is crucial in the schema matching process. The correct assignation of meaning (concerning the semantic source) to every schema label enables the possible discovery of accurate semantic correspondences among the elements of different schemas. The major contribution of this paper is the proposal of an approach to resolve semantic ambiguous problems and representation of the intended meaning of schema labels according to the context in which they appear in a non-redundant way. The CKBD method integrates semantic domain and more frequency use—two kinds of context knowledge—into a disambiguation processor.

Meanwhile, a set of concepts that are part of the domain makes up the context knowledge. From the results, it can be concluded that the CKBD approach is capable of resolving ambiguous words, and semantic ambiguity in structured and semi-structured, which consists of multiple words. This paper significantly contributes to the body of knowledge in this field via the proposal of a new technique that resolves lexical ambiguity and semantic ambiguity in structured and semi-structured data posed to a schema integration system. The method proposed in this study enhances the usability of heterogeneous data sources. It also reduces semantic conflicts and efficiently solves lexical ambiguity.

REFERENCES

[1] Hossain, J., Sani, N.F.M., Affendey, L.S., Ishak, I., And Kasmiran, K.A.: 'Semantic Schema Matching Approaches: A Review,' Journal of Theoretical & Applied Information Technology, 2014, 62, (1)

[2] Tahat, S., and Ahmad, K.: 'Semi-automated schema integration (case): A tool to identify and resolve naming conflicts,' Australian Journal of Basic & Applied Sciences, 2013, 7

[3] Ahmad, K., Chiew, H.K., and Samad, R.: 'Intelligent Schema Integrator (ISI): A tool to solve the problem of naming conflict for schema integration,' in Editor (Ed.)^(Eds.): 'Book Intelligent Schema Integrator (ISI): A tool to solve the problem of naming conflict for schema integration' (IEEE, 2011, edn.), pp. 1-5

[4] Ahamed, B.B., Ramkumar, T., and Hariharan, S.: 'Data integration progression in the large data source using mapping affinity,' in Editor (Ed.)^(Eds.): 'Book Data integration progression in the large data source using mapping affinity' (IEEE, 2014, edn.), pp. 16-21

[5] Blomqvist, E., and Thollander, P.: 'An integrated dataset of energy efficiency measures published as linked open data,' Energy Efficiency, 2015, 8, (6), pp. 1125-1147

[6] Nicklas, D., Schwarz, T., and Mitschang, B.: 'A Schema-Based Approach to Enable Data Integration on the Fly,' International Journal of Cooperative Information Systems, 2017, 26, (01), pp. 1650010

[7] Kettouch, M.S., Luca, C., Hobbs, M., and Fatima, A.: 'Data integration approach for semi-structured and structured data (Linked Data),' in Editor (Ed.)^(Eds.): 'Book Data integration approach for semi-structured and structured data (Linked Data)' (IEEE, 2015, edn.), pp. 820-825

[8] He, W., and Da Xu, L.: 'Integration of distributed enterprise applications: A survey,' IEEE Transactions on Industrial Informatics, 2014, 10, (1), pp. 35-42

[9] Díaz, M., Martín, C., and Rubio, B.: 'State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing,' Journal of Network and Computer Applications, 2016, 67, pp. 99-117

[10] Bergamaschi, S., Beneventano, D., Po, L., and Sorrentino, S.: 'Automatic normalization and annotation for discovering semantic mappings': 'Search computing' (Springer, 2011), pp. 85-100

[11] Bilke, A.: 'Duplicate-based Schema Matching,' 2007

[12] Rahm, E., and Bernstein, P.A.: 'A survey of approaches to automatic schema matching,' the VLDB Journal, 2001, 10, (4), pp. 334-350

[13] Alwan, A.A., Nordin, A., Alzeber, M., and Abualkishik, A.Z.: 'A Survey of Schema Matching Research using Database Schemas and Instances', International Journal Of Advanced Computer Science And Applications, 2017, 8, (10), pp. 102-111

[14] Nguyen, Q.V.H., Nguyen, T.T., Miklos, Z., Aberer, K., Gal, A., and Weidlich, M.: 'Pay-as-you-go reconciliation in schema matching networks,' in Editor (Ed.)^(Eds.): 'Book Pay-as-you-go reconciliation in schema matching networks' (IEEE, 2014, edn.), pp. 220-231

[15] Gillani, S., Naeem, M., Habibullah, R., and Qayyum, A.: 'Semantic schema matching using DBpedia,' International Journal of Intelligent Systems and Applications, 2013, 5, (4), pp. 72

[16] Rachman, M.A.F., and Saptawati, G.A.P.: 'Database integration based on combination schema matching approach (case study: Multi-database of district health information system),' in Editor (Ed.)^(Eds.): 'Book Database integration based on combination schema matching approach (case study: Multi-database of district health information system)' (IEEE, 2017, edn.), pp. 430-435

[17] Bhattacharjee, S., and Ghosh, S.K.: 'Automatic resolution of semantic heterogeneity in GIS: An ontology-based approach': 'Advanced Computing, Networking and Informatics-Volume 1' (Springer, 2014), pp. 585-591

[18] Bellström, P.: 'Schema Integration: How to Integrate Static and Dynamic Database Schemata' (Karlstads Universitet, 2010. 2010)

[19] Unal, O., and Afsarmanesh, H.: 'Schema matching and integration for data sharing among collaborating organizations', Journal of Software, 2009, 4, (3), pp. 248-261

[20] Unal, O., and Afsarmanesh, H.: 'Using linguistic techniques for schema matching,' in Editor (Ed.)^(Eds.): 'Book Using linguistic techniques for schema matching' (2006, edn.), pp. 115-120

[21] WSD, W.S.D.: 'Word Sense Disambiguation,' 2015

[22] Al-Harbi, O., Jusoh, S., and Norwawi, N.: 'Handling ambiguity problems of natural language interface for question answering,' International Journal of Computer Science Issues (IJCSI), 2012, 9, (3), pp. 17

[23] Po, L.: 'Improving data integration through disambiguation techniques,' Lecture Notes in Computer Science, 2008, 5039, pp. 372-375

[24] Stevenson, M., and Wilks, Y.: 'Word sense disambiguation,' The Oxford Handbook of Comp. Linguistics, 2003, pp. 249-265

[25] Ponzetto, S.P., and Navigli, R.: 'Knowledge-rich word sense disambiguation rivaling supervised systems,' in Editor (Ed.)^(Eds.): 'Book Knowledge-rich word sense disambiguation rivaling supervised systems' (Association for Computational Linguistics, 2010, edn.), pp. 1522-1531