ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ENHANCED NORMALIZATION APPROACH ADDRESSING STOP-WORD COMPLEXITY IN COMPOUND-WORD SCHEMA LABELS

[1]JAFREEN HOSSAIN, [2]NOR FAZLIDA MOHD SANI, [3]LILLY SURIANI AFFENDEY, [4]ISKANDAR ISHAK, [5]KHAIRUL AZHAR KASMIRAN

Faculty of Computer Science and Information Technology, Universiti Putra Malaysia

E-mail: [1]jafreen@gmail.com, [2]fazlida@upm.edu.my, [3]lilly@upm.edu.my, [4]iskandar_i@upm.edu.my, [5]k_azhar@upm.edu.my

ABSTRACT

An extensive review of the existing schema matching approaches discovered an area of improvement in the field of semantic schema matching. Normalization and lexical annotation methods using WordNet have been somewhat successful in general cases. However, in the presence of stop-words these approaches result in poor accuracy. Stop-words have previously been ignored in most studies resulting in false negative conclusions. This paper proposes NORMSTOP (NORMalizer of schemata having STOP-words) as an improved schema normalization approach that addresses the complexity of stop-words (e.g. 'by', 'at', 'and,' or') in Compound Word (CW) schema labels. Using a combined set of WordNet features, NORMSTOP isolates these labels during the preprocessing stage and resets the base-form to a relevant WordNet term, or an annotable compound noun. When tested on the same real dataset used in the earlier approach - (NORMS or NORMalizer of Schemata), NORMSTOP shows up to 13% improvement in annotation recall measurement. This level of improvement takes the overall schema matching process another step closer to perfect accuracy; while its absence exposes a gap in expectation, especially in today's databases, where stop-words are in abundance.

Keywords: *Database Integration, Schema Matching, Data Heterogeneity, Semantic Schema Matching, Schema Label Normalization, Stop-Words*

1. INTRODUCTION

The advancement of information and communication technology has opened doors for many data sources to communicate with each other in a semantic web. At the same time, it has created data heterogeneity problems in various application domains. Large amount of data is created every day by different sources in different formats. The value of data increases when it can be linked with other data, thus successful data integration is a major creator of value. Data integration and data sharing are hence getting more important for many application domains. At the same time, the semantic integration is getting crucial and complex due to this large scale of data and its heterogeneous nature. This heterogeneity can be in terms of data source format, types, representation, or semantic interpretation. The schema matching problem is considered by many researchers as one of the bottlenecks for semantic integration. It is not a new research area and has received increasing attention since the 1970s [10, 11]. Numerous matching approaches, strategies and algorithms have been developed.

Currently, the schema matching process has improved from fully manual to semi-automatic after years of studies by many researchers. Automatic or semi-automatic schema matching has to deal with problems arising from the heterogeneity of data sources which can be distinguished into two main types of heterogeneity: structural and semantic heterogeneity [4, 12]. Structural heterogeneity means differences among attribute types, formats, or models whereas semantic heterogeneity means differences in the meaning or annotation of schema elements. In this paper, we have mainly focused on semantic heterogeneity and one of its aspects.

It has been proven that schema normalization approaches improve the lexical relationship and matching accuracy among schema labels. Lexical annotation (i.e. annotation with reference to a lexical resource/dictionary, e.g. WordNet) helps to relate a “meaning” to schema labels. However, the accuracy of semi-automatic lexical annotation methods on real life schemas still suffer from the problem of non-dictionary words such as compound words (CWs), abbreviations and acronyms.

Schema normalization approaches can help to resolve this problem and increase the number of similar schema labels. But it gets even more complicated when the compound words have stop-words in them. Stop-words have been mostly ignored in previous studies of schema matching and hence the schemas having stop-words have missed out any annotation.

The objective of the research work is to propose an approach for solving the problem of stop-words in schema labels and improve the lexical annotation of schema label normalization by reducing false negative (ie. missing right annotation) results. So, in this paper the NORMSTOP approach has been introduced which takes stop-words into annotation consideration, changes schema into relevant WordNet word form and improves the Normalization Approach of NORMS. [15]

The study concentrates on the Compound Word (CW) annotation, which includes Compound Nouns (CNs), or Compound Word formats containing stopwords and the consequent false negative (missing a right annotation) problem. Not all the stop-words used in natural language processing (NLP) has been considered in this study since only some common stop-words are used in database designing. The main focus of the research is on stop-words found in the test dataset [22, 23]. Those are “in”, “by”, “at”, “to”, "from", “on”, “since”, “upto”, “until”, “till”, “is”, “are”, “was”, “were”, “or” respectively.

In order to fulfill the above mentioned objective, we assume that a fully functional schema normalization tool is implemented and available, in which we can add and run the newly developed algorithm.

Schema matching is an important and essential process in different domains including e-commerce, data-integration, health-care and many more. By identifying the stop-word in compound word schema labels, the proposed approach would reduce the false negative results in schema normalization and annotation process which is an integral part of schema matching.

Section 2 focuses on the specific problem of stopwords in schema label which is the main focus of this research work. Section 3 of this document discusses the relevant literature review and some previous approaches on schema and ontology matching. It also details schema normalization approaches and NORMS (NORMalizer of Schemata), an existing tool to perform schema label normalization to enhance the automatic result of schema matching process and some open problems of this area. Section 4 states the methodology to solve the problem mentioned in section 2 and also discusses the new proposed approach — NORMSTOP and its step by step procedures. Section 5 focuses on explaining the implementation of the proposed approach. Section 6 details out the evaluation of its results in comparison with previous NORMS approach. Section 7 discussed the results using different data sets, section 8 focuses on some limitation of NORMS approach and section 9 displayed some real-life implication of NORMSTOP whereas section 10 concludes the thesis, mentioning the main contribution and discusses of NORMSTOP and some future opportunities in the same domain.

2. PROBLEM DISCUSSION

As mentioned by Sorrentino et al. [4, 15], the weakness of a thesaurus, like WordNet, is that it does not always cover the detail information of a specific domain and domain-dependent terms or words, or non-dictionary words (such as Compound words, abbreviations, acronyms etc). So, this kind of non-dictionary words in schema labels strongly affects the automatic lexical annotation technique. To address this problem, they presented a method for schema label normalization which expands abbreviations and automatically annotates Compound Nouns (CNs) by enriching WordNet with new meanings.

With regards to the schema label normalization method, Sorrentino et al. [15] mentioned some limitation and future improvements in their work which would take into consideration the main problem during the experimental evaluation: The presence of stop-words (e.g. “to”, “at”, “and” etc.) in schema labels; and the problem of false negative (ie. missing right annotation) non-dictionary words during the identification steps of schema normalization [15].

Po and Sorrentino [2, 15] also stated the recall rate was affected by the existence of non-endocentric (endocentric CNs is a kind of CNs consisting of a head and modifier) CNs (such as “ManualPublished”, “isMember” or

“InProceedings”) in the schemas for all the data sets and that their method could not identify.

So, the limitations summarized from [15] that were not considered while processing schema label normalization are:

1) Other kinds of multi-word units (e.g. prepositional verbs such as “WrittenBy”)
2) The use of conjunctions (such as “and” or “or”) in schema and ontology labels
3) The presence of stop-words (e.g. “to”, “at”) in schema and ontology labels

Considering the limitations mentioned by Sorrentino et al. [15], one specific problem has been identified summarizing the three problems stated above which needs improvement:

Problem of the presence of stop-words (e.g. “to”, “at”, “and” etc.) in schema labels resulting false negative lexical annotation during schema normalization process [15]

3. LITERATURE REVIEW

The aim of recent research work on integrating web databases has been to allow uniform access to the large amount of data behind query interfaces. Source discovery, query interface extraction, schema matching are some of the tasks among the many in this integration process [7]. There are some other important tasks that are commonly ignored or assumed to be solved either manually or by other system. Finding the set of stop-words and its semantic relationship with other words within the label is one such task. It has a very important and direct impact on determining the meaning specifically the synonymy, hyponymy or any such potential relationship between labels.

Data-cleaning, within the scope of integration, is considered by previous researcher as a significant and integral part, when multiple data sources need to be integrated, e.g., in data warehouses, federated database systems or global web-based information systems [16, 17]. It was mentioned that to facilitate instance matching and integration, attribute values should be converted to a consistent and uniform format. For example, date and time entries should be brought into a specific format; names and other string data should be converted to either upper or lower case, etc. Text data may be condensed and unified by performing stemming, removing prefixes, suffixes, and stop words. Furthermore, abbreviations and encoding schemes should consistently be resolved by consulting special synonym dictionaries or applying predefined conversion rules.

Most of the traditional schema matching techniques also considers stop-words as noise and removes them at the preprocessing step. In COMA [1, 6, 8], it was mentioned that multi-word element names are often composed of multiple words, such as DeliveryAddress, DevAddr, and AddrOfDel. Directly comparing those names unlikely yields correct similarity due to order and abbreviation of the words in the names. Hence, different pre-processing techniques, such as tokenization (DeliveryAddress → {delivery, address}), removing of stop-words (such as of), stemming (delivery → deliver) applied to obtain a set of the most characteristic words for comparison.

MOMIS [3] implemented an automatic annotation algorithm which includes stemming and stop words removal functionalities in order to optimize the annotation phase and increase the annotation accuracy.

Table 1: Different Schema Matching Preprocessing steps handling stop-words

Schema Matching System	**Author/ Year**	**Preprocessing Steps**		
		Tokenization	Stop-word Removal	Stemming
COMA	(Do and Rahm, 2002) [6]	√	√	√
MOMIS	(Bergamaschi et al., 1998) [3]	√	√	√
SeMap	(Wang et al., 2006) [18]	√	√	√
RiMOM	(Li et al., 2009) [13]	√	√	√
Ontology Matching	(Hlaing et al., 2009) [9]	√	√	√
large scale schema matching	(He and Chang, 2004) [8]	√	√	√
Cupid	(Madhavan et al, 2001) [14]	√	√	√

Hlaing et al., [9], mentioned the preprocessing process as (1) Tokenization : break an item into atomic words e.g., break “fromCity” into “from” and “city” ,break “first_name” into “first” and “name”, (2) Expansion words : expand abbreviations and acronyms to their full words, e.g., “dept” to “departure” (3) Stop word removal and stemming : “the model” into “model” (remove a, an, of, the, etc) and (4) Standardization of words – Irregular words are standardized to a single form, e.g., “colour” to “color”. For normalization, it exploits and uses domain specific dictionary. This dictionary consists all existing terms in a specific domain include their synonym sets.

In SeMap [18], schema matching system mentioned that it eliminates those frequently used words that can be found in a list. Li et al., [13], also mentioned tokenization, stop-word removal and stemming as the pre-processing step before performing the schema matching task.

Table 1, listed different studies where tokenization, stop-word removal and stemming process have been used before doing schema matching.

A widespread review has been conducted on several previous schema matching approaches, strategies and techniques till recent times. We ended up focusing on semantic schema matching approaches and its significance in the overall schema matching process. It can be concluded from this review that the implicit meaning or semantics of schema labels plays an important role in the exercise of discovering mappings between different data sources.

Although many strategies were developed to solve this problem including schema normalization approaches (Sorrentino et al., 2011) it was obvious there is still room for improvement and future work. List of such future work included finding the meaning of domain specific terms, different compound words having prepositional-verbs, conjunctions, digits, or stop-words in schema labels. Also, more work can be done to improve the number of false positive and false negative relationships. Another relevant future research could possibly be the inclusion of instance-based matching techniques to improve the automatic annotation and relationship discovery processes among schema labels.

Schema label normalization is an imperative step in the whole process of schema matching. In absence of a proper schema normalization process, schema matching results in poor accuracy, due to abundance of false negative and false positive matching. The schema label normalization process used in NORMS pulls off an impressive success in covering most common aspects of abbreviation expansion and CN annotation. However, the normalization in NORMS fails to cover some of the commonly used labels in database systems in recent times, due to the presence of stop-words.

After going through the problem regarding stop-words in different domains, like information retrieval, data mining, text mining in natural language processing, data integration, it has been noticed that stop-words are considered as noise and filtered out in most the cases to save processing time and to get quick result. But, while performing data integration in schema normalization process for schema labels, the stop-words need to stay, rather than being filtered out, in order to understand the actual meaning of the schema label.

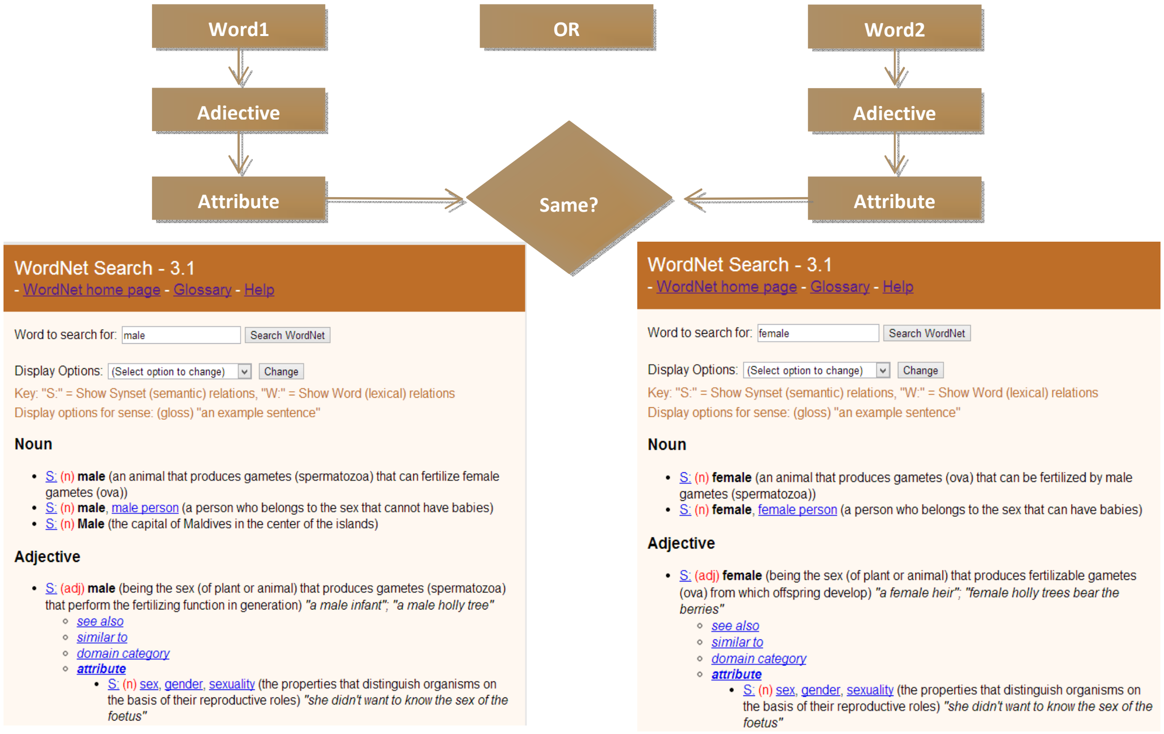
4. METHODOLGY

In the first phase, this study proposes an approach called “NORMSTOP” in order to resolve the “stop-word” problem in schema normalization approach. It aims to be an improved approach compared to NORMS, the schema normalizer approach, developed by Sorrentino et al. [4, 15]. With NORMSTOP, the overlaying framework has been revised for the normalization approach, and an underlying algorithm has been developed to specifically solve the stop-word problem.

While designing NORMSTOP, two main objectives were focused – i) to improve overall success rate of NORMS lexical annotation by eliminating the false negative results generated due to the presence of stop-words in schema labels and ii) To ensure that we do not deteriorate the existing NORMS results with changes introduced by NORMSTOP.

Based on 1) the observation of limitation in the anchor paper [15], 2) careful analysis of 15 real datasets, and 3) further studies on stop-words, the following prepositions, conjunctions, and auxiliary verbs were selected to be covered in this study [5]:

Prepositions: “by”, “at”, “to”, “from”, “on”, “in”, “since”, “upto”, “till”, “until” ; Conjunctions: “or”; Auxiliary Verb: “is”, “was”, “are”, “were”.



WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: male Search WordNet

Display Options: (Select option to change) Change
Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations
Display options for sense: (gloss) "an example sentence"

Noun

- S: (n) **male** (an animal that produces gametes (spermatozoa) that can fertilize female gametes (ova))
- S: (n) **male**, male person (a person who belongs to the sex that cannot have babies)
- S: (n) **Male** (the capital of Maldives in the center of the islands)

Adjective

- S: (adj) **male** (being the sex (of plant or animal) that produces gametes (spermatozoa) that perform the fertilizing function in generation) *"a male infant"; "a male holly tree"*
  - *see also*
  - *similar to*
  - *domain category*
  - ***attribute***
    - S: (n) sex, gender, sexuality (the properties that distinguish organisms on the basis of their reproductive roles) *"she didn't want to know the sex of the foetus"*

WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: female Search WordNet

Display Options: (Select option to change) Change
Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations
Display options for sense: (gloss) "an example sentence"

Noun

- S: (n) **female** (an animal that produces gametes (ova) that can be fertilized by male gametes (spermatozoa))
- S: (n) **female**, female person (a person who belongs to the sex that can have babies)

Adjective

- S: (adj) **female** (being the sex (of plant or animal) that produces fertilizable gametes (ova) from which offspring develop) *"a female heir"; "female holly trees bear the berries"*
  - *see also*
  - *similar to*
  - *domain category*
  - ***attribute***
    - S: (n) sex, gender, sexuality (the properties that distinguish organisms on the basis of their reproductive roles) *"she didn't want to know the sex of the foetus"*

Figure 1: Searching same attribute for opposing words in WordNet

Future improvement of this study can expand to other stop-words that have not been covered in this research.

With the above stop-words in consideration, the overlaying framework of NORMS was revised (Figure 3) to include checkpoints for non-WordNet schema labels that qualify as compound words having stop-words.

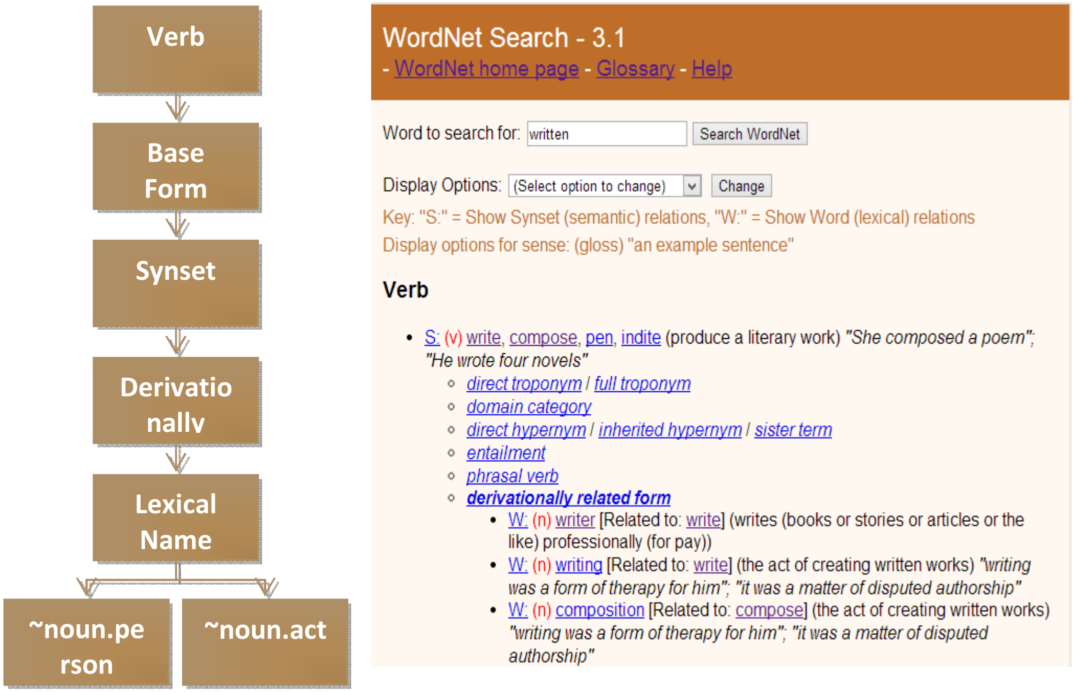
These labels were then normalized, within the underlying NORMSTOP algorithm, using combination of various features available from WordNet. Some of the key features that have been used were: 1) Derivationally Related Forms, 2) Attributes, and 3) LexNames [24].

1) Derivationally Related Forms (DRF) were used to discover possible act and actor forms of a verb, post fixed by preposition for time or agent. For example, "writing" and "writer" are DRFs of the word "write". This feature helped normalize labels like "writtenBy" to "writer", and helped address most of the prepositional stop-word problems.

2) Attributes have been used to find out the common concept among two opposing objectives separated by an "or" stop-word. For example, "sex" is the common attribute for adjectives "male" and "female". Figure 1 shows this procedure. And that's how the "attributes" feature from WordNet helped address the conjunction ("or") stop-word problems.

3) LexNames were used to identify the appropriate act and actor forms from the DRFs. LexNames for "writing" and "writer" are "noun.act" and "noun.person" respectively. Figure 2 on next page shows the procedure to find noun form of a verb from WordNet.

In case of auxiliary verbs, the main task was to identify if the schema was a table name or an attribute name – the normalization later is done accordingly.



Word to search for: written

Display Options: (Select option to change)

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Verb

- S: (v) write, compose, pen, indite (produce a literary work) *"She composed a poem"; "He wrote four novels"*
  - direct troponym / full troponym
  - domain category
  - direct hypernym / inherited hypernym / sister term
  - entailment
  - phrasal verb
  - **derivationally related form**
    - W: (n) writer [Related to: write] (writes (books or stories or articles or the like) professionally (for pay))
    - W: (n) writing [Related to: write] (the act of creating written works) *"writing was a form of therapy for him"; "it was a matter of disputed authorship"*
    - W: (n) composition [Related to: compose] (the act of creating written works) *"writing was a form of therapy for him"; "it was a matter of disputed authorship"*

Figure 2: Changing verb to noun using DRF in WordNet

5. IMPLEMENTATION

NORMSTOP was implemented with relatively small modification to the existing source code of NORMS. However, the framework design of the algorithm was critical to the success of this NORMSTOP implementation.

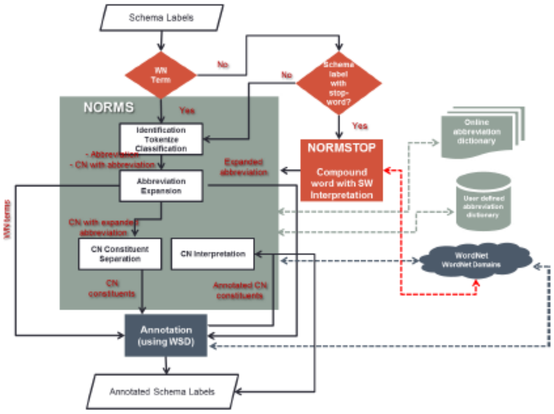
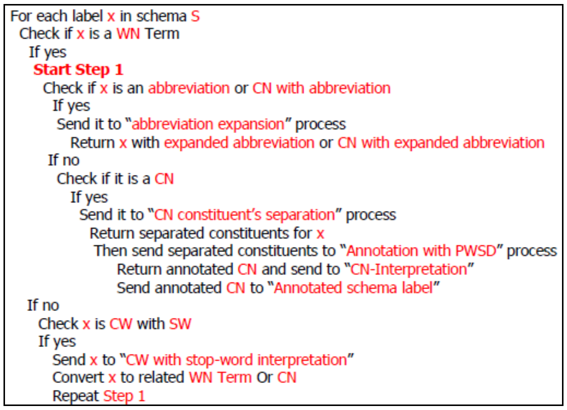

Schema Labels
WN Term
No
Schema label with stop-word?
No
NORMS
Yes
Identification Tokenize Classification
- Abbreviation
- CN with abbreviation
Abbreviation Expansion
Expanded abbreviation
NORMSTOP
Compound word with SW Interpretation
Yes
Online abbreviation dictionary
User defined abbreviation dictionary
WN terms
CN with expanded abbreviation
CN Constituent Separation
CN Interpretation
WordNet WordNet Domains
CN constituents
Annotated CN constituents
Annotation (using WSD)
Annotated Schema Labels

Figure 3: NORMSTOP Framework

The figure above shows the framework of NORMSTOP. It has two main additional procedures on top of NORMS to handle the stop-word complexity. The processes are: i) Qualify for NORMSTOP, ii) NORMSTOP. The Overlaying Frame Algorithm: The Complete Annotation Process describes all three (including NORMS) of these processes and their relationship.

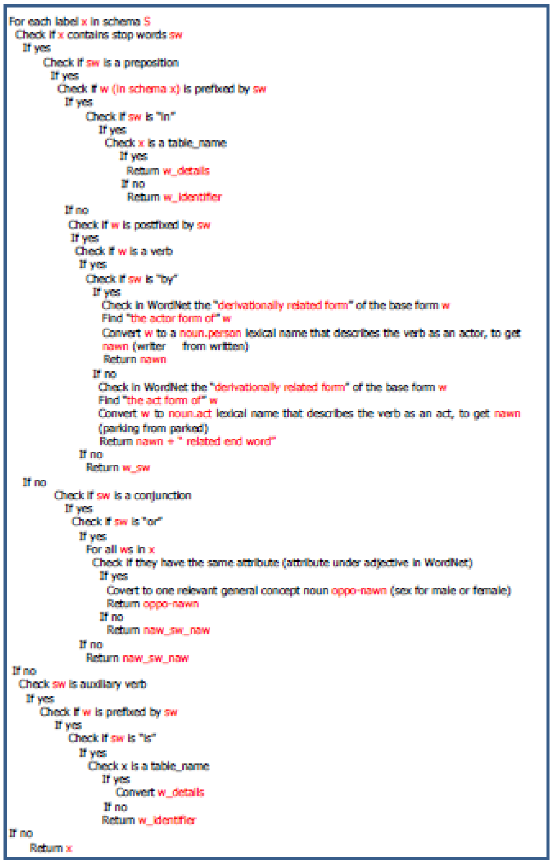


```
For each label x in schema S
  Check if x is a WN Term
    If yes
     Start Step 1
      Check if x is an abbreviation or CN with abbreviation
       If yes
        Send it to "abbreviation expansion" process
          Return x with expanded abbreviation or CN with expanded abbreviation
      If no
        Check if it is a CN
          If yes
           Send it to "CN constituent's separation" process
             Return separated constituents for x
              Then send separated constituents to "Annotation with PWSD" process
                Return annotated CN and send to "CN-Interpretation"
                Send annotated CN to "Annotated schema label"
    If no
     Check x is CW with SW
     If yes
       Send x to "CW with stop-word interpretation"
       Convert x to related WN Term Or CN
       Repeat Step 1
```

Figure 4: The Overlaying Frame Algorithm

The Underlying Focal Algorithm: CW with SW Interpretation is the heart of this study. This algorithm explains how the implementation of the NORMSTOP process has been achieved. Variables used in this algorithm are the following:

x : Schema label
S : Schema
st : Stop-words
naw : Normalized and annotated word
nawn : Normalized and annotated word converted to Noun
oppo-nawn : Two opposing words converted to one conceptual noun



```
For each label x in schema S
 Check if x contains stop words sw
  If yes
     Check if sw is a preposition
      If yes
       Check if w (in schema x) is prefixed by sw
        If yes
           Check if sw is "in"
            If yes
             Check x is a table_name
               If yes
                Return w_details
               If no
                Return w_identifier
        If no
         Check if w is postfixed by sw
          If yes
           Check if w is a verb
            If yes
             Check if sw is "by"
              If yes
               Check in WordNet the "derivationally related form" of the base form w
               Find "the actor form of" w
               Convert w to a noun.person lexical name that describes the verb as an actor, to get
               nawn (writer    from written)
               Return nawn
              If no
               Check in WordNet the "derivationally related form" of the base form w
               Find "the act form of" w
               Convert w to noun.act lexical name that describes the verb as an act, to get nawn
               (parking from parked)
               Return nawn + " related end word"
            If no
             Return w_sw
  If no
       Check if sw is a conjunction
        If yes
         Check if sw is "or"
          If yes
           For all ws in x
            Check if they have the same attribute (attribute under adjective in WordNet)
             If yes
              Covert to one relevant general concept noun oppo-nawn (sex for male or female)
              Return oppo-nawn
             If no
              Return naw_sw_naw
          If no
           Return naw_sw_naw
If no
 Check sw is auxiliary verb
  If yes
    Check if w is prefixed by sw
      If yes
        Check if sw is "is"
          If yes
           Check x is a table_name
             If yes
               Convert w_details
             If no
             Return w_identifier
If no
   Return x
```

Figure 5: Underlying Focal Algorithm

As seen above, the algorithms have focused a lot on re-using the NORMS functions and ensuring that the default results (true positive and true negatives) from original NORMS were not allowed to be deteriorated.

6. TEST AND EVALUATION

The main objective of the evaluation method is to measure the performance of NORMSTOP approach in comparison with NORMS (NORMalizer of Schemata) [15] and to check whether NORMSTOP approach improves the lexical annotation process and thus improve the overall schema matching process.

Precision (P), Recall (R) and F-measure are used in the research work as the performance measurement tool. To measure the effectiveness of NORMSTOP, Gold Standard lexical annotation has been manually generated with the help of a human expert.

Precision and Recall address the quality of the results automatically determined by the annotation tool. In Figure 6, the set of derived annotations is comprised of B, the true positives, and C, the false positives. False negatives (A) are annotation needed but not automatically annotated, while false positives are annotations falsely identified by the tools. True negatives, D, are annotation failures, which have also been correctly failed by the automatic annotation process. Obviously, both false negatives and false positives reduce the match accuracy.

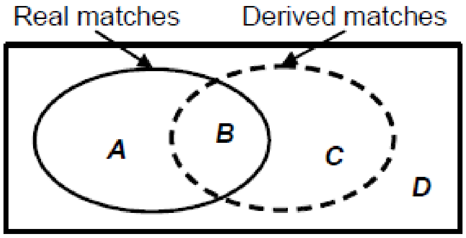


Figure 6: Underlying Focal Algorithm

Based on the cardinality of these sets, two common measures, Precision (P) and Recall (R), which originate from the information retrieval field, can be calculated as follow:

$$Precision(P) = \frac{|B|}{|B| + |C|}$$

$$Recall\ (R) \ = \frac{|B|}{|A| + |B|}$$

We have Precision = Recall = 1, when no false negatives and false positives are returned, in ideal situation. However, neither Precision nor Recall alone can perfectly evaluate the match quality. Precisely, recall can be maximized at the expense of a poor Precision. On the other hand, a high Precision can be derived at the value of a low recall by returning only few (correct) annotations (Karasneh et al., 2010). The weighted harmonic of these two measures precision and recall is F-measure, which is the measure of performance that takes into account both Precision and Recall. The F-measure is calculated as follows: where P is the Precision measure and R is the Recall measure.

$$F - measure = \frac{2}{\frac{1}{R} + \frac{1}{P}}$$

The results obtained in both NORMS and NORMSTOP have then been compared with respect to the corresponding Gold Standard. This study uses the same data as has been used by NORMS.

The following three data sets: (1) OAEI, (2) Mondial, and (3) Amalgam (an integration benchmark for bibliographic data) [19] have been used in the evaluation process as these have been used in the anchor paper. Each data set consists of two schemas that are to be merged. These data sets have been used by other matching systems as well [20][21]. These data sets are particularly appropriate to assess schema normalization as they consist of several non-dictionary words, represent various application domains; finally contains both relational, XML schemas and ontologies.

For each schema element in each dataset, the NORMS and NORMSTOP results were ranked as one of the 4 possible outcomes: false positive (FP), true positive (TP), false negative (FN)‖ or true negative (TN). In order to identify the right ranking, the annotation (both NORMS and NORMSTOP) were checked against the Gold standard annotation. Once all the schema elements NORMS annotation results are identified as one of the 4 possible outcomes, all the 4 ranking are then summed up. The summed-up quantities are then used to identify precision and recall value for NORMS annotation. F-measure is then calculated using the precision and recall values. The exact same is done based on the total count of NORMSTOP rankings and thus the precision, recall and F-measure of NORMSTOP is also retrieved.

7. RESULTS AND DISCUSSION

The performance of NORMS complemented with NORMSTOP was measured with precision, recall and F-measures. These results were compared against the default NORMS measurement for the same. In order to measure the given KPIs (Key Performance Indicator), we had used expert opinion from database administrators to come up with the set of gold standards, which we took as the set of real lexical annotation for the 3 real data sets. We, then compared results from both the default and improved NORMS against this golden standard and scored each of the numerous schema labels in each dataset as TP, FP, TN or FN based on true or false and positive or negative outcome. Each dataset is then scored for both NORMS and NORMSTOP and the results speak for themselves.

Amalgam dataset had a total of 189 schema labels, out of which only 6 were compound words with stop-words. So, the chances of improvement through the NORMSTOP approach was only limited to these 6 schema labels. NORMSTOP was able to improve the results on the Amalgam dataset only marginally. It reduced the total number of false negatives from 7 to 5, and converted these 2 items to true positive. This improvement resulted in an overall 2% improvement in recall and 1% improvement in F-measure. NORMSTOP managed to retain the original NORMS success intact, on top of achieving this new improvement. There was no deterioration seen in precision, which in turn reflects the improvement in F-measure. In fact, the precision improves a little due to the improvement in true positive results.

OAEI had 55 schema labels, which were compound word with stop-words. In other words out of the total 455 schema labels, more than 10% schema labels had stopwords in them. So, this dataset was a perfect example of a real dataset that shows the necessity of a tool like NORMSTOP, in order to do a successful lexical annotation. the result on OAEI dataset from NORMSTOP shows a staggering 13% improvement in recall and an equally impressive 7% improvement in F-measure. The new approach managed to reduce as many as 52 (60%) false negative, and converted all of them to true positive results. The new approach managed to retain the NORMS success and made sure that no successful results from NORMS (True positives and True negatives) were deteriorated with the introduction of NORMSTOP.

Mondial dataset had a total of 276 schema labels, out of which only 6 were compound words with stop-words. Similar to Amalgam, this dataset had only 6 schema labels where NORMSTOP could contribute to improve the results. NORMSTOP managed to make a small improvement of 1% in recall and less than 1% improvement in F-measure. It reduced the total number of false negatives from 7 to 4, which actually was better than Amalgam in volume;

however, considering the bigger total dataset size this didn't even surpass Amalgam results. However, even in this dataset NORMSTOP managed to retain the original NORMS success harmless.

As with any other research, we have tried to isolate and identify any negative impact NORMSTOP might have contributed, while trying to improve the accuracy of lexical annotation. While we know NORMSTOP is not capable of solving all relevant problems of lexical annotation, it is also to be appreciated that NORMSTOP does a decent job in retaining the positive results of NORMS approach, without resulting in any worsening of NORMS results. The known limitations and possible future work for NORMSTOP has been detailed out in the next section.

As far as significance is concerned, we believe 13% improvement in accuracy of lexical annotation can be considered as an important step forward for the field of lexical annotation and hence normalization and semantic schema matching. With more focus on the listed improvement areas of this research can contribute to substantial progress to the field of this study.

In summary, it was observed that with the introduction of NORMSTOP, the overall result of NORMS lexical annotation improves by up to 13%. The other noteworthy achievement of this new approach is the successful retention of positive results from the previous effort. It is often observed that newer approach and the consequent success only comes at the cost of partial deterioration to the previous results. However, NORMSTOP manages to avoid that completely, while still resulting in a significant improvement. We believe these results can be improved even further by taking care of the limitations described in the next section.

8. LIMITATIONS OF NORMS APPROACH

Sorrentino et al. (2011), has mentioned several limitations about the NORMS tool they have developed for the schema normalization task. Some limitations are listed below:

- Other kinds of multi-word units (e.g. prepositional verbs such as "WrittenBy") other than endocentric compound nouns

- The use of conjunctions (such as "and" or "or") in schema and ontology labels

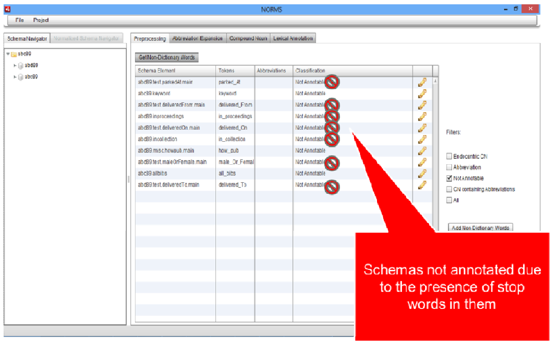
- The presence of stop words (e.g. "to", "at") in schema and ontology labels

- A different kind of non-dictionary words, i.e. words which are not present in a lexical resource as they belong to a specific application domain (e.g. medicine, architecture or biology)

- Digits in schema labels

- Problem of false negative non-dictionary words during the identification step (e.g. "RID", "AID")

We have developed a test data set to check the limitations of stop-words in schema label and found out that NORMS tool classifies them as not-Annotable‖. Figure 7 below shows the same limitations.



Schema Element	Tokens	Abbreviations	Classification
abd89.test.parkedAt.main	parked_At		Not Annotabl
abc89.keyword	keyword		Not Annotable
abd89.test.deliveredFrom.main	delivered_From		Not Annotabl
abd89.inproceedings	in_proceedings		Not Annotabl
abd89.test.deliveredOn.main	delivered_On		Not Annotabl
abd89.incollection	in_collection		Not Annotabl
abd89.misc.howpub.main	how_pub		Not Annotable
abd89.test.maleOrFemale.main	male_Or_Femal		Not Annotabl
abc89.allbibs	all_bibs		Not Annotable
abd89.test.deliveredTo.main	delivered_To		Not Annotabl

□ Endocentric CN
□ Abbreviation
☑ Not Annotable
□ CN containing Abbreviations
□ All

Schemas not annotated due to the presence of stop words in them

Figure 7: Limitations of NORMS, showing "Not Annotable" On Test Data

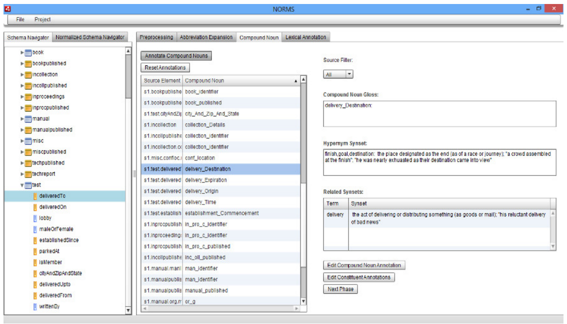
9. REAL LIFE IMPLICATION OF NORMSTOP

Table 2 shows the shortlisted stop-words, their categories, probable position in compound-word and the result after going through NORMSTOP.

Table 2: Shortlisted stop-words in NORMSTOP

Type of stop-	Stop-words	Position in CW	Example	
			CW with SW	Result
Preposition	In (table name)	Prefix	InProceeding	Procceding_Details
	In (attribute	Prefix	InProceeding	Proceeding_Identifier
	By	Postfix	WrittenBy	Writer
	At	Postfix	ParkedAt	Parking_Location
	To	Postfix	DeliveredTo	Delivery_Destination
	On	Postfix	DeliveredOn	Delivery_Time
	In	Postfix	DeliveredIn	Delivery_Place
	From	Postfix	DeliveredFrom	Delivery_Origin
	Since	Postfix	EstablishedSince	Establishemnet_ Commencement
	Upto	Postfix	DeliveredUpto	Delivery_Expiration
	Until	Postfix	ValidUntil	Validity_Expiration
	Till	Postfix	ValidTill	Validity_Expiration
Conjunction	Or	Anywhere	MaleOrFemale	Sex
Auxiliary Verb	Is (table name)	Prefix	IsMember	Member_Details
	Is (attribute	Prefix	IsMember	Member_Identifier

Figure 8 shows the snapshot from improved NORMS tool showing result. If we notice carefully, it can be seen that the tool showing a test table which is showing the attribute name “deliveredTo” and its conversion to “delivery_Destination” after the normalization process through NORMSTOP in improved NORMS.



Source Element	Compound Noun
s1.bookpublishe	book_identifier
s1.bookpublishe	book_published
s1.test.cityAndZi	city_And_Zip_And_State
s1.incollection	collection_Details
s1.incollpublishe	collection_identifier
s1.incollection.c	collection_identifier
s1.misc.confloc.i	conf_location
s1.test.delivered	delivery_Destination
s1.test.delivered	delivery_Expiration
s1.test.delivered	delivery_Origin
s1.test.delivered	delivery_Time
s1.test.establish	establishment_Commencement
s1.inprocpublish	in_pro_c_identifier
s1.inproceeding	in_pro_c_identifier
s1.inprocpublish	in_pro_c_published
s1.incollpublishe	inc_oll_published
s1.manual.manl	man_identifier
s1.manualpublis	man_identifier
s1.manualpublis	manual_published
s1.manual.org.n	or_g

Compound Noun Gloss:
delivery_Destination:

Hypernym Synset:
finish,goal,destination: the place designated as the end (as of a race or journey); "a crowd assembled at the finish"; "he was nearly exhuasted as their destination came into view"

Related Synsets:

Term	Synset
delivery	the act of delivering or distributing something (as goods or mail); "his reluctant delivery of bad news"

Figure 8: Short-listed stop-word category in NORMSTOP

10. CONCLUSION AND FUTURE WORK

This study signifies the critical role of semantic schema normalization in the vastly experimented field of schema matching and discovers the necessity of addressing the issues arisen from the presence of stop-words in schema labels often used by the many database designers in today’s world of big data. With a planned strategic algorithm that takes leverage from the existing research on NORMS, this study introduces NORMSTOP that makes a worthwhile contribution to schema normalization approaches in general and towards a future evolution of database integration.

NORMS had already successfully accomplished the major task of semantic schema label normalization taking into consideration the complexities of compound noun and abbreviation. However, it does not cover the next level of complications in a typical schema label, which is related to compound words having stop-words. This study has focused on addressing these important limitations in NORMS and has taken it to the next level, where accuracy of schema normalization has significantly improved.

For one of the real datasets used in the study, NORMSTOP showed a significant recall improvement of 13%. OAEI had a total of 455 schema labels. NORMS had failed to annotate a total of 88 labels in that particular dataset, out of which 55 were due to stop-words. NORMSTOP managed to improve 52 of these 55 false negatives, and successfully annotated them to result in true positive outcome, matching accurately with manual expert annotation. Moreover, NORMSTOP managed to retain the true positives results from NORMS, resulting in no deterioration in precision, and an overall significant (7%) improvement in F-measure.

With the limitations sorted out in future, this can yield in even better results. For example, expanding from only WordNet to other auxiliary resources like – ConceptNet, Wikipedia, DBpedia, and Google search might help reduce the limitations around multi-domain complexity.

As regard to schema label normalization method, future work might include the problem of identifying digits in schema labels. Also, the problem around false negative non-dictionary words during the identification step (e.g. “RID”, “AID” etc) can be improved. In NORMSTOP method, more stop-words identifier can be added for future requirement, for example, “Of”, “With”, “How” etc.

Annotating schema labels, keeping its data type in consideration, was not implemented in NORMSTOP. For example, the schema label “parkedAt” can mean both time and place, depending on its data type. When its data type is text or similar, most likely it means “parking_place”. However, when its data type is date/time it definitely means “parking_time”. In future improvement of NORMSTOP this separation can be a certain addition.

This version of NORMSTOP also fails to annotate schema label such as” MOrF”, which can mean “MaleOrFemale” or “Gender”. By integrating to NORMS‘abbreviation expansion function and with re-structuring of NORMSTOP code, this can be improved in the next release.

Inclusion of instance-based matching techniques might also improve the overall schema label normalization process. Sometimes, schema labels contain misleading elements which is difficult to normalize without considering the data type of the instances. For example, the meaning of the label “phone” is “electronic equipment” according to WordNet, but in real situation it should refer to phone numbers.

Instance based technique might also help to annotate the non-informative schema labels. It is common practice of many database designers to use code while naming the column of a table, for example, DB_FB02, which is difficult to normalize and annotate automatically even for a schema designer. So, using the instance based technique will help reduce this problem.

Gradual minimization of the gaps identified above can definitely contribute to an evolution towards an almost perfect and fully automated normalization approach in future. NORMSTOP has taken that evolution one step closer to reality and continued efforts can only mean that the world is ready to embrace the big data explosion.

11. ACKNOWLEDGEMENT

The authors acknowledge the financial supported by the Fundamental Research Grant under grant numberFRGS/1/2014/ICT07/UPM/02/2 received from the Ministry of Higher Education, Malaysia via Universiti Putra Malaysia (UPM).

REFRENCES:

[1] Aumueller, D., Do, H. H., Massmann, S., and Rahm, E., (2005). Schema and ontology matching with COMA++. In Proc. of Special Interest Group on Management of Data, SIGMOD’05, New York, NY, USA, pages 906–908. ACM.

[2] Bergamaschi, S., Po, L., and Sorrentino, S. (2008). Automatic annotation for mapping discovery in data integration systems, in SEBD, S. Gaglio, I. Infantino, and D. Sacca`, Eds., pp. 334–341.

[3] Bergamaschi, S., Castano S, Vincini, M. (1998). MOMIS, An Intelligent System for the Integration of Semi Structured and Structured Data, INTERDATA.

[4] 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, Springe.

[5] http://www.semantikoz.com/blog/free-stop-word-lists-in-23-languages/

[6] 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.

[7] Dragut, E., Fang, F., Sistla, P., Yu., C., Meng, W., (2009). Stop Word and Related Problems in Web Interface Integration, VLDB Endowment, ACM.

[8] He, B., Chang, K., C., C. (2004). A holistic paradigm for large scale schema matching. SIGMOD Rec., 33(4):20–25.

[9] Hlaing, S. (2009). Ontology based Schema Matching and Mapping Approach for Structured Databases.ICIS, November 24-26, Seoul, Korea.

[10] Islam, A., Inkpen, D. (2008). Semantic text similarity using corpus-based word similarity and string similarity, ACM Transactions on Knowledge Discovery from Data, Vol.2, No.2, Article 10.

[11] Islam, A., Inkpen, D. Z., Kiringa, I.(2008). Applications of corpus-based semantic similarity and word segmentation to database schema matching. VLDB J., 17(5):1293–1320.

[12] 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.

[13] Li, J., Tang, J., Li, Y., Luo, Q. (2009). RiMOM: A Dynamic Multistrategy

[14] Ontology Alignment Framework. IEEE Trans. Knowl. Data Eng. 21(8), 1218-1232.

[15] 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.

[16] Proc. of the 27th International Conference on Very Large Data Bases (VLDB 2001), September 11-14, 2001, Roma, Italy, pages 49–58. Morgan Kaufmann.

[17] Po, L., Sorrentino, S. (2011). Automatic generation of probabilistic relationships for improving schema matching. Information Systems Journal, Special Issue on Semantic Integration of Data, Multimedia, and Services, 36(2):192208.

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

[19] Rahm E, Do, H. H. (2000). Data Cleaning: Problems and Current Approaches. University of Leipzig, Germany.

[20] Wang, T. (2006). SeMap: a generic schema matching system. University of British Columbia.

[21] Miller, R. J., Fisla, D., Huang, M., Kalmuk, D., Ku, F., and Lee, V. (2001). The Amalgam Schema and Data Integration Test Suite.

[22] The Clio Project. (2007, July 10), Retrieved from http://dblab.cs.toronto.edu/project/clio/index.php#testschemas

[23] Ontology Alignment Evaluation Initiative. (2009, May 6), Retrieved from http://oaei.ontologymatching.org/2008/results/benchmarks/

[24] JWI 2.4.0. (2007-2013), Retrieved from http://projects.csail.mit.edu/jwi/api/