International Journal of Advances in Computer Science and Its Applications– IJCSIA
Volume 4: Issue 2 *[ISSN: 2250-3765]*
Publication Date : 25 June 2014

Utilizing WordNet for Instance-based Schema Matching

Ahmed Mounaf Mahdi and Sabrina Tiun

Abstract— **Instance-based matching is the process of identifying the correspondences of schema elements by comparing the instances of different data sources. It is used as an alternative option when the schema-based matching fails. Instance-based matching is applied in many application areas such as website creation and management, data warehousing, database design, and data integration. Many recent approaches focus on instance-based matching. In this paper, we propose an approach that utilizes WordNet-based measure for string domain by getting the similarity coefficient in the range of [0..1]. In previous approach, the regular expression is achieved with a good accuracy for numerical instances only and is not implemented on string instances because we need to know the meaning of string to decide if there is a match or not. The using of WordNet-based measures for string instances should guarantee to improve the effectiveness in terms of Precision (P), Recall (R) and f-measure (F). In this paper we implemented Lin's measure to find the similarity of two instances. This approach is evaluated with real dataset and the results are found better than using just equality measure for string especially if the schemas are disjoint. The approach achieved 91.8% f-measure (F).**

Keywords— **Schema matching, Instance-based matching, WordNetm similarity measures, Lin's measure.**

I. Introduction

A relational schema is the logical definition of an entity that includes the entity name and a set of elements with their data types. When these relational schemas collected together, the concept of database will appear. Database schema is a structure of database that describes the arrangement of its instances, relationships and constraints [1].

Schema matching is a process of identifying the semantic correspondences between elements of the many database schemas [1, 2, 3, 4]. See figure 1 for the mapping elements of two schemas. Schema matching finds the similarity or the semantic relationships between elements of two schemas existing in different data repositories. Solving this problem is very important in many applications such as schema integration, website creation and management, schema migration, database design, data warehousing, and data integration.

The existing approaches in schema matching classified into three levels: (1) Schema level which is using structural schema information; (2) Instance level which is using a stored data instances; (3) Hybrid which combines information from schema structure and stored instances [5]. Sometimes, the schema information (element name, data type, description, etc) is not available or is not possible to get the correct matching, especially when the element name is abbreviation, therefore, if the schema matching failed, the focus will be on values stored in the schemas. For these reasons, many recent approaches focus on instance-based matching [7, 8].

Instance-based matching is needed in many applications, such as data and schema integration. Suppose two companies decided to corporate with each other; in this case, they need to integrate their databases. As it is known that every company has documents stored in the databases with different schemas and to integrate these schemas, the detecting of matched candidates is needed for the merging process [9].

Most approaches in instance-based schema matching [10, 12] used the similarity metrics to measure the similarity between elements and detect the match if exists. Mehdi el al. [7] used the regular expression (regex) to find the correspondences of elements. The process of instance matching using regular expression achieve with a good accuracy for numerical and mixed data instances because the data can be described using a specific pattern, but it is not possible to apply the regex on string domain. The previous approach [7] used the regex for matching numerical instances only, while for the elements with the string data type, a tokenizing process is implemented by considering the first token only for each instance. This will generate a problem of detecting the match of non-match strings such as *hot dog* will match *hot*. In addition, it will not match the instances that have the same meaning, such as *car* will not match *automobile* and also for cities, such as *Los Angeles* will not match *New York* [7, 12].

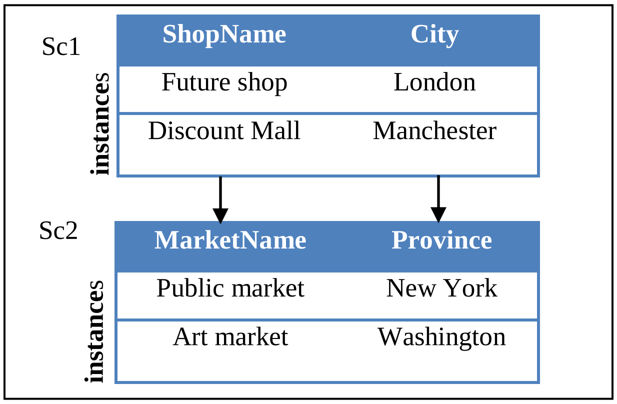
This paper uses Lin's measure to find the similarity between the instances of string elements to generate candidates of correspondence elements. Lin's measure relies on WordNet to get the similarity coefficient in the range of 0 and 1. Recently, many concerns have been put on semantic similarity measures that depend on WordNet [6, 13, 14, 15].

The rest of this paper arranged as follow: Section (II) discusses the techniques that are used to find the similarity of terms, and the previous works on instance-based schema matching. Section (III) discusses the WordNet-based measure and explains about Lin's measure in details. The

Ahmed Mounaf Mahdi
Faculty of Information Science & Technology / UKM University Malaysia

Sabrina Tiun
Faculty of Information Science & Technology / UKM University Malaysia

proposed approach is discussed in details in section (IV). Section (V) shows the experiment results that have been conducted to evaluate our approach, and finally section (VI) concludes our approach and highlights the new ideas that will be accomplished as a future works.



ShopName	City
Future shop	London
Discount Mall	Manchester

MarketName	Province
Public market	New York
Art market	Washington

Figure 1 Mapping elements of two schemas.

II. Literature Review

Most of previous works [10, 12, 16, 17, 18, 19] have used similarity metrics techniques to find the similarity of instances. These metrics have been classified into two categories: (i) character-based and (ii) token-based similarity measures.

Character-based measure is useful for typographical errors and useless for recognizing the rearrangement of terms such as *data analyzing* and *analyzing data*. This measure is elaborated as Edit distance, Jaro distance and Q-gram. Edit distance metric is used by Levenshtein [20], the measure depends on the number of edit operations, insert, delete, and replace characters that transformed the string into another string. Jaro distance metrics depends on the number of common characters in the two strings [21]. Finally, Q-gram metric depends on the sequence of N characters for comparing two strings. For example, *Prov* and *Provider* are similar because they shared two trigram *pro*, *rov* as follow: *pro*, *rov* for *Prov* and *pro*, *rov*, *ovi*, *vid*, *ide*, *der* for *Provider*[22].

Token-based similarity is useful for recognizing the rearrangement of terms and is implemented by breaking the strings into substrings. Jaccard, Atomic strings, and Cosine similarity are examples of token-based similarity. Jaccard [23] proposed a technique to compute the similarity of distributions of Flora in distinct geographical areas. Monge and Elkan [24] proposed a technique to match two atomic strings. The matching between the two atomic strings will be success if they are equal or one of the two strings is a prefix of the other. The number of matching atomic strings can be divided by the average number of atomic strings to find the similarity between two elements [25]. Cosine similarity, this technique solved the problem of recognizing the rearrangement of terms that mentioned above, so it can consider the words *analyzing data* and *data analyzing* are similar. The drawback of this similarity measure is the limitation of solving the spelling error issues. For example, *deta analyzing* will not be similar to *analyzing data* [25].

Mehdi et al. [7] proposed an approach to find the correspondences of instances by using regular expression. Their approach generates the regex list automatically and finds the correspondence of numeric and mixed columns by matching the regex which is generated for a specific column in first schema, with the columns of second schema. For the string values the authors take a first token of the attribute to compare with the first token of strings existing in the other schema. They depend on equality measure of two strings regardless the meaning of these strings. The approach has been evaluated with series of experiments and the results achieved 98% accuracy which is better than other string similarity metrics that the authors compared with, such as; LCS, which its accuracy is 95.90%.

Zapilko et al. [11] also utilized the regular expression to solve the matching problem. The authors define a list of regex which describe a different element such as a purely numeric element or mixed data element. The approach is achieved a good result for matching numeric and mixed instances only.

Zaiß et al. [12] also used the regular expression but in ontology matching area, their approach is relied on instances for matching ontologies. They scan the instances to describe the contents of every instance using a set of regular expressions. Each of regular expression represents a concept. The regex list is created by a domain expert to fit the instances of ontologies for a specific domain. The comparison process will be started by comparing the regex list with the instances, when the regular expression is matched with an instance, the regex will be assigned to the instance. After that the regex which is assigned to the most of instances that belong to a specific element is consider as regex for this element. For creating the candidate mapping, cosine similarity measure is implemented to find 1:1 mapping in order to find the highest similarity of two instances.

Yatskevich and Giunchiglia [6] proposed an approach that utilize WordNet as a knowledge source for getting the semantic relations of two concepts instead of similarity coefficient with values [0..1]. The authors present twelve element level matchers which utilize WordNet to get the semantic relation. They evaluated their approach with other matching systems and the results were comparable with 42% precision (P) and 58% recall(R).

Bilke and Naumann [26] have proposed a new algorithm that finds the duplicates in a tuple, so they utilize a duplicate detection for instance-based schema matching problem. In their approach, the matching task will be on duplicate records that are identified during the first step of their approach. For duplicate detection they used cosine measure that relies on assignment of weights for each token in every tuple. These weights represent the importance of token in tuple. After that, the authors presented a similarity function that can identify duplicates on partially overlapping schema. At the end, after duplicate records detected and the similarity matrix built, the matching on the matrix will be performed by using the matrix as an input to bipartite weighted matching problem which is known as assignment problem [27]. The optimal solution of this problem is the maximum summation of similarities. The authors implemented six experiments and they achieved from 90% to 100% precision (P) and from 95% to 100% recall(R).

Dhamankar et al. [28] proposed iMAP system for 1-1 matching as well for complex matching. This approach considers the matching task as a search in very large match space. For this reason, they utilize a set of searchers and each searcher can detect a specific type of elements. The authors used specialized searchers which are: (i) text searcher, for concatenation of text elements, (ii) numeric searcher for collecting elements with arithmetic data and (iii) date searcher for some elements that their data are date. Beam search [29] is used for managing the search to get a set of matches which are reranked using name similarity of elements. The last step is to select the best candidates by considering domain knowledge and the compatibility of constraints. iMAP finds the matching of relational schemas only but as the authors said that it is possible to use this idea for other models. This approach helps human to interact with the system to find the matched elements quickly. The experiment results on numerous real datasets achieved about only 43% to 92% accuracy.

III. WordNet-based Measures

There are many techniques used to find the similarity or relatedness between two concepts. We have found that the character-based similarity measures and token-based similarity measures are not suitable for matching if there are no shared characters between the two comparing concepts.

The problem of finding the correctly matched elements is not a trivial to be solved because of structure variety and semantic diversity of data. Some auxiliary sources such as dictionary and thesauri can help to reduce the degree of difficulty [30].

In recent years, several concerns have been put on measuring based on WordNet [6, 13, 14, 15]. For this reason we utilized the WordNet in this paper to help us to find the similarity between two concepts.

WordNet is the product of research project that performed at Princeton University [31]. WordNet includes three databases; the first is for nouns, the second is for verbs and the third for adjectives and adverbs. WordNet also includes a set of synonyms which are also called synsets. A synset represents a concept or a sense of a set of terms. Synsets produce different semantic relationships such as *synonymy* which is the similar relationship and *antonymy* which is the opposite relationship, *hypernymy*/ *hyponymy* which are super concept/sub concept relationship also called Is-A hierarchy / taxonomy, *meronymy* which is part-of relationship and *holonymy* which is has-a relationship. The semantic relations through the synsets are varies depending on the grammatical category. WordNet also produces some descriptions of each concept (gloss) including definitions and examples.

Semantic similarity measures are used for implementing some tasks such as term disambiguation [32], text segmentation [33], and for consistency of ontologies. Many measures have been proposed, all measures are categorized by Meng et al. [13] into four categories: path length-based measures, information content-based measures, feature-based measures, and hybrid measures. In the next subsection, we will discuss about one of the content-based measures which is Lin's measure. We implemented the Lin's measure in this paper to find the similarity of instances to obtain the matched elements.

A. *Information Content-based Measure*

This measure considers that every concept has a lot of information in WordNet. Similarity measures are relying on the information content of the concept. If there is much common information between the concepts, then the two concepts have the same meaning. Lin [34] proposed a similarity measure that uses both the information content that subsumes the concepts in taxonomy and the information needed to fully describe these concepts. The similarity values of this measure are ranged between 0 and 1.

$$\mathrm{Sim}_{\mathrm{Lin}}(c_{1,}c_2) = (2*\mathrm{IC}(\mathrm{lso}(c_1,c_2))) / (\mathrm{IC}(c_1)+\mathrm{IC}(c_2)) \qquad (1)$$

Where IC is an information content and $\mathrm{lso}(c_1,c_2)$ is the lowest common subsummer.

There is no standard to evaluate the effectiveness of semantic similarity measures. If a specific application requires a measure of semantic similarity, we have to implement the measure to find the performance of using the measure in a specific area[13].

WordNet::Similarity[1] is a software package developed at the University of Minnesota as open source software for Perl. It helps the user to find the semantic similarity or the relatedness between two concepts. This system provides six similarity measures and three relatedness measures based on the WordNet database [35]. The similarity measures are based on is-a hierarchy. These measures are divided into only two groups path-based measures and information content based measures, however it does not include feature-based measure. For our approach, we used WordNet Similarity For Java[2] (WS4J), which provides a Java API of Princeton's English WordNet. It is a re-implementation of Wordnet::similarity for Perl that mentioned above.

IV. The Proposed Approach

The framework of our proposed approach is organized into the following steps:

[1] http://www.d.umn.edu/~tpederse/similarity.html

[2] https://code.google.com/p/ws4j/

- Prepare the dataset.
- Identifying the data type for each column to find out whether a specific column is a string, or not.
- Select the samples randomly from each column.
- Perform the matching with WordNet if the data type is string.
- The final output will be the matched elements.

The first step in our approach is reading the schemas and storing the instances in array after that, analysis the data to determine the string elements. The third step is selecting the samples; in our approach we selected 10% of the total number of records for comparing process. The fourth step is to calculate the similarity of a selected instance from a specific element in schema A with instances from all elements in schema B. If the similarity value is more than a predefined threshold, the elements of these instances are consider as a candidate of match. We depend on Lin’s measure to find the similarity. The choosing of threshold is depended on series of experiments. In our approach the best result is obtained when the threshold set to 0.76.

We have built a function that calculates the similarity of two items (S_1,S_2). The items are the current item from the source schema and every string item of the target schema.

The items are sent as a one token(S_1,S_2) to compare a compound words that has a specific meaning as a one concept such as some cities like *Los Angeles* and also we sent the items as a list of tokens (tokens(S1),tokens(s2)). For example, if S_1 is *Los Angeles* and S_2 is *New York*, the system will calculate the similarity of *Los Angeles* with *New York* will find a high similarity, *Los* with *New* will not find a similarity, *Los* with *York* will not find a similarity, *Los* with *New* will not find a similarity, *Angeles* with *New* will not find a similarity, *Angeles* with *York* will not find a similarity. Another example if S_1 is *American* and S_2 is *American new*, the calculation will be: *American* with *American new* will not find a match, *American* with *American* will find a match, *American* with *New* will not find a match.

In CalcWordNet algorithm which is illustrated in the figure 2, *lin* function calculates the similarity of two terms that comes from two elements by using Lin’s measure. This function is a part of set of functions included in WS4J API. If the value of similarity is more than a predefined threshold, the possibility of mapping the two elements together will increase by one in an array *degree* as in line 14 in figure2. The maximum value of *degree* will consider that the element (i) and element (j) as a matched elements.

```
Algorithm : CalcWordNet
  1. Pass In: S1: represents a token from Schema 1
  2.          S2: represents a token from Schema 2
  3.           Pos: represent the part of speech for these tokens.
  4.           i: represents index of the current column of schema 1
  5.           j: represents index of the current column of schema 2
  6. Pass Out: degree array
  7. Let sense ←1 which is the most common sense.
  8. Let degree be a two dimensional array for the similarity degree
  13. IF lin(S1,sense,S2,sense,pos)≥0.76 then
  14.     degree[i][j]←degree[i][j]+1
  15. End IF
```

Figure 2. CalcWordNet algorithm.

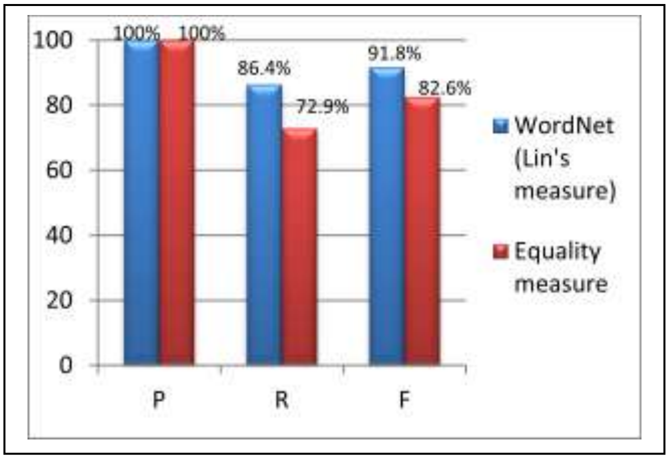


Figure 3. Comparison of using Lin’s measure or equality measure for string domain.

V. Experiment Result

The experimental results of using Lin’s measure which use WordNet as their central resource have been accomplished for string domain to evaluate the performance of our proposed approach. These experiments aim: (i) to measure the effectiveness of the match results based on Precision (*P*), Recall (*R*) and F-measure (F). (ii) to highlight the performance of our proposed approach and compare it with equality measure that is used by [7] for string domain. We have implemented Lin’s measures and equality measure using restaurant dataset which is available online[3].

As it is illustrated in figure 3 the Precision (P) of using Lin’s measure or equality measure is 100%. This means the set of wrong mapping is small while, the use of Lin’s measure has increased the recall (R) with 13.5% differences. Consequently, using of Lin’s measure achieved 91.8% f-measure (F) and using of equality measure has achieved only 82.6% f-measure (F).

VI. Conclusion and Future Work

We conclude from the experiment result that the use of Lin’s measure is a good choice and better than using just equality measure. Lin’s measure increased the f-measure (F) with 9.2%.

The choosing of threshold value directly affects on the effectiveness of our approach and the best result we got it after series of experiments when we set the threshold at 0.76.

As a future works, more experiments have to be conducted on different datasets. In addition, we intend to implement other WordNet-based measures to find the best measure.

[3] http://www.infochimps.com/datasets/restaurant

We need to combine this measure (Lin's measure) that is used for string domain with other techniques such as regular expression technique for numeric and mixed domain. This should guarantee to increase the effectiveness of instance-based matching approach.

References

[1] Gillani, S., M. Naeem, R. Habibullah & A. Qayyum 2013. Semantic Schema Matching Using DBpedia. *International Journal of Intelligent Systems and Applications (IJISA)* 5(4): 72.

[2] Li, W.-S. & C. Clifton 1994. Semantic integration in heterogeneous databases using neural networks. *VLDB*, hlm. 12-15.

[3] Milo, T. & S. Zohar 1998. Using schema matching to simplify heterogeneous data translation. *VLDB*, hlm. 24-27.

[4] Madhavan, J., P. A. Bernstein & E. Rahm 2001. Generic schema matching with cupid. *Proceedings of the International Conference on Very Large Data Bases*, hlm. 49-58.

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

[6] Yatskevich, M. & F. Giunchiglia (2004) Element level semantic matching using WordNet.Meaning Coordination and Negotiation Workshop, ISWC, hlm.

[7] Mehdi, O. A., H. Ibrahim & L. S. Affendey 2012. Instance based Matching using Regular Expression. *Procedia Computer Science* 10: 688-695.

[8] Gomes de Carvalho, M., A. H. Laender, M. André Gonçalves & A. S. Da Silva 2012. An evolutionary approach to complex schema matching. *Information Systems.*

[9] Shvaiko, P. & J. Euzenat. 2005. A survey of schema-based matching approaches. Dlm. (pnyt.). Ed. Journal on Data Semantics IV hlm. 146-171. Springer.

[10] Tejada, S., C. A. Knoblock & S. Minton 2002. Learning domain-independent string transformation weights for high accuracy object identification. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, hlm. 350-359.

[11] Zapilko, B., M. Zloch & J. Schaible 2012. Utilizing Regular Expressions for Instance-Based Schema Matching.

[12] Zaiß, K., T. Schlüter & S. Conrad 2008. Instance-Based Ontology Matching Using Regular Expressions. *On the Move to Meaningful Internet Systems: OTM 2008 Workshops*, hlm. 40-41.

[13] Meng, L., R. Huang & J. Gu (2013) A Review of Semantic Similarity Measures in WordNet. International Journal of Hybrid Information Technology 6(1).

[14] Varelas, G., E. Voutsakis, P. Raftopoulou, E. G. Petrakis & E. E. Milios (2005) Semantic similarity methods in wordNet and their application to information retrieval on the web. Proceedings of the 7th annual ACM international workshop on Web information and data management, hlm.10-16.

[15] Lin, F. & K. Sandkuhl (2008) A survey of exploiting wordnet in ontology matching. Dlm. (pnyt.). Ed. Artificial Intelligence in Theory and Practice II hlm. 341-350. Springer.

[16] Tejada, S., C. A. Knoblock & S. Minton 2001. Learning object identification rules for information integration. *Information Systems* 26(8): 607-633.

[17] Bilenko, M., R. Mooney, W. Cohen, P. Ravikumar & S. Fienberg 2003. Adaptive name matching in information integration. *Intelligent Systems, IEEE* 18(5): 16-23.

[18] Duchateau, F., Z. Bellahsene & M. Roche 2006. A context-based measure for discovering approximate semantic matching between schema elements.

[19] Rong, S., X. Niu, E. W. Xiang, H. Wang, Q. Yang & Y. Yu. 2012. A machine learning approach for instance matching based on similarity metrics. Dlm. (pnyt.). Ed. *The Semantic Web–ISWC 2012* hlm. 460-475. Springer.

[20] Levenshtein, V. I. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet physics doklady*, hlm. 707.

[21] Jaro, M. A. 1989. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *Journal of the American Statistical Association* 84(406): 414-420.

[22] Moreau, E., F. Yvon & O. Cappé 2008. Robust similarity measures for named entities matching. *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, hlm. 593-600.

[23] Jaccard, P. 1912. The distribution of the flora in the alpine zone. 1. *New Phytologist* 11(2): 37-50.

[24] Monge, A. E. & C. Elkan 1996. The field matching problem: Algorithms and applications. *Proceedings of the second international Conference on Knowledge Discovery and Data Mining*, hlm. 267-270.

[25] Elmagarmid, A. K., P. G. Ipeirotis & V. S. Verykios 2007. Duplicate record detection: A survey. *Knowledge and Data Engineering, IEEE Transactions on* 19(1): 1-16.

[26] Bilke, A. & F. Naumann 2005. Schema matching using duplicates. *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, hlm. 69-80.

[27] Papadimitriou, C. & K. Steiglitz 1988. Combinatorial Optimization, Mineola, NY, DOVER PUBLICATIONS, INC.

[28] Dhamankar, R., Y. Lee, A. Doan, A. Halevy & P. Domingos 2004. iMAP: discovering complex semantic matches between database schemas. *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, hlm. 383-394.

[29] Russell, S. J., P. Norvig, J. F. Canny, J. M. Malik & D. D. Edwards. 1995. Artificial intelligence: a modern approach Ed. 74. Prentice hall Englewood Cliffs.

[30] Liang, Y. 2008. An instance-based approach for domain-independent schema matching. *Proceedings of the 46th Annual Southeast Regional Conference on XX*, hlm. 268-271.

[31] Miller, G. & C. Fellbaum 1998. Wordnet: An electronic lexical database, MIT Press Cambridge.

[32] Patwardhan, S., S. Banerjee & T. Pedersen. 2003. Using measures of semantic relatedness for word sense disambiguation. Dlm. (pnyt.). Ed. *Computational linguistics and intelligent text processing* hlm. 241-257. Springer.

[33] Kozima, H. 1994. Computing lexical cohesion as a tool for text analysis.Tesis Citeseer.

[34] Lin, D. 1998. An information-theoretic definition of similarity. *ICML*, hlm. 296-304.

[35] Fellbaum, C. 1998. A semantic network of english: the mother of all WordNets. *Computers and the Humanities* 32(2-3): 209-220.

About Author (s):

Ahmed Mounaf Mahdi is a Master student of Information Technology (Computer Science) in UKM University, Malaysia. His research interests include schema matching, pattern recognition and semantic similarity.

Sabrina Tiun is currently a senior lecturer at the Universiti Kebangsaan Malaysia.
Her research work and interests range from Natural Language processing, Speech Processing to Information Retrieval.