BOHR International Journal of Intelligent Instrumentation and Computing
2021, Vol. 1, No. 1, pp. 5–9

https://doi.org/10.54646/bijiiac.002
www.bohrpub.com

Measure Term Similarity Using a Semantic Network Approach

D. M. Kulkarni and Swapnaja S. Kulkarni

Department of Computer Science Engineering, D.K.T.E. Society's Textile and Engineering Institute, Ichalkaranji, Maharashtra, India

Abstract. Computing semantic similarity between two words comes with variety of approaches. This is mainly essential for the applications such as text analysis, text understanding. In traditional system search engines are used to compute the similarity between words. In that search engines are keyword based. There is one drawback that user should know what exactly they are looking for. There are mainly two main approaches for computation namely knowledge based and corpus based approaches. But there is one drawback that these two approaches are not suitable for computing similarity between multi-word expressions. This system provides efficient and effective approach for computing term similarity using semantic network approach. A clustering approach is used in order to improve the accuracy of the semantic similarity. This approach is more efficient than other computing algorithms. This technique can also apply to large scale dataset to compute term similarity.

Keywords: Term similarity, Multi-word expression, Clustering, Semantic network.

1 Introduction

Semantic similarity measurement is the fundamental problem. Computation between two terms mainly appears in lexical semantics [1]. Here similarity between two terms can be measure. The term in the sense a single word or multi-word expression can be taken. This technique of computing semantic similarity between words can be used in many applications such as in case of web search or in document search [2]. In wed there are thousands of data is available in a very large scale. These data from web can be used to compute term similarity. Two terms are semantically similar if they have some common attributes. For example "apple" and "company". These two terms are semantically similar because both terms belong from the same category. Both terms are companies. For example "car" and "journey". These two terms are not semantically similar but both are related. Because "journey" is an activity and "car" is a transport mean for "journey".

WordNet [3] is a dataset consists of thousands of words. It maintains isA relation between words. Two terms are considered as semantically similar if there is having isA relation present between two terms. That's why semantic similarity is hard to model as compare semantic relatedness. There are two main approaches to compute semantic similarity between two terms. These approaches are: 1. knowledge based 2. corpus based approaches.

In knowledge based approach most of the work in this space [4] depends on isA relation between words in a WordNet. isA relation between words is mandatory to compute semantic similarity between words. Another one is corpus based approach little bit different from knowledge approach. In corpus based approach contexts of a term can be extracted from a large scale dataset. In short this work is mainly related with web. Here a corpus can be anything from a webpage or web search snippet. Here terms are extracted from web search engines to compute semantic similarity.

But there are some limitations faced by knowledge based approach. The main problem with this is a limitation of taxonomy with WordNet. This approach is not able to cover all senses of terms.WordNet does not consists of all word sense pairs. Instead of that it may contains only single word with phrase of multi-word expressions. This is impossible to compute semantic similarity between unknown terms and its senses in WordNet.

Corpus based approach is having some limitations. In this approach semantic similarity can be computed by using search engines. Search engine uses indexing and ranking mechanisms for words. There is one limitation that user must know exactly what they are searching for. Otherwise it may give ambiguous results for it. For example if user searches for an "apple". Then search engine may give all possible results of an "apple" such as apple as fruit, apple as company. It may generate an ambiguity. To deal

with this approach user should clear there concept regarding terms to compute semantic similarity.

This system proposes an efficient and effective approach for computing semantic similarity between words. isA relation is present between words to compute similarity. Depends on their relation similarity score of terms can be decided. After completion of similarity computation similarity score can be generated. It generates similarity score a number between 0 and 1. A system uses such a dataset which is having isA relation between two terms.

This system is more reliable and efficient to compute semantic similarity between two terms because clustering approach is introduced. Refined approach algorithm is introduced to accurately compute semantic similarity between words. This system is also able to solve problems with ambiguous in meaning.

In this paper, we propose an efficient and effective framework for computing semantic similarity (a number between 0 and 1) between two terms using a large scale, general purpose isA network obtained from a web corpus. Below is a small sample of results:

- High similarity (synonyms): hgeneral electric, gei Synonyms that refer to the same entity should have the highest similarity score.
- High similarity (ambiguous terms): hmicrosoft, applei, horange, redi Words such as “apple” and “orange” have multiple senses. However, when people compare “apple” with “microsoft”, they consider “apple” in the sense of a company rather than a fruit, and when they compare “orange” and “red”, they consider “orange” as a color rather than a fruit. Thus, disambiguation needs to be performed by default in similarity comparison.
- Low similarity (though share same hypernyms in WordNet): hmusic, lunchi , hbanana, beefi These pairs of terms are not similar. However, in an isA network, “music” and “lunch” may both belong to concepts such as “activity”, and “banana” and “beef” may both belong to concepts such as “food". We may use their distances in a handcrafted taxonomy to measure similarity, but handcrafted taxonomies have low coverage, while distances in large scale, data driven semantic networks are not easy to measure.

2 Literature Survey

A new semantic relatedness measurement using wordnet features [5] by M. A. H. Taieb, M. B. Aouicha, and A. B. Hamadou system introduces a fundamental problem of computing semantic similarity between two terms. Information Content (IC) method is used here to compute similarity between words. This method also used a taxonomical feature between terms. This approach is having two parts: subgraph is formed in first part. Its descendents are count as compartmentalization parameters. In second part IC metric is integrated into multistrategy approach.

This system Using information content to evaluate semantic similarity in a taxonomy [8] by P. Resnik, introduces an isA taxonomy to compute semantic similarity between two terms. Here information content (IC) method is used to compute similarity. This method is same as edge counting method. The results of this system show that it produces sensible results using IC technique.

This system Exploring knowledge bases for similarity [9] by E. Agirre, M. Cuadros, G. Rigau, and A. Soroa introduces graph based algorithms to compute similarity. For computing similarity it uses WordNet along with graph based algorithms. Wordnet353 dataset is used to compute similarity between words. This system is more better than other traditional systems. Results show that it gives performance improvement as compare to traditional system.

3 Problem Statement

To generate similarity score from given datasets implement basic approach and refined approach algorithm for computing semantic similarity between pair of words.

4 Proposed Work

The above diagram illustrates architecture of semantic similarity. Where a user may perform login and may give a query to the database. Here admin is responsible to maintain a dataset. After generating a dataset it may upload that into database. From a database terms are extracted and given as an input to type checking method. Then context of a term can be extracted from its type. By using a clustering algorithm according to its contexts clusters are formed [12]. Finally by using similarity functions semantic similarity can be measured and output is generated.

The proposed system is designed and developed with following modules, which are given as below:

Module 1: Candidate Set of Words From Data Dictionary

A dataset may consist of collection of words. It consists of more than 100,000 words, where words may have multiple definitions. It may contain phrases P. P is nothing but word or sequence of words. Words in dataset can be relate with each other by its type such as dataset may consists of synonyms set, antonyms set, hypernym set, hyponym set of words. For maintaining a dataset an algorithm is used which takes sequence of terms as an input. And the output of this method is set of words. If a required dataset is available then by using that one semantic similarity can be

measured. Otherwise by using algorithms dataset can be generated.

Module 2: Type Checking

A first step while computing semantic similarity is to check the type of given term. Type of a term can be either an entity or a concept. For type checking of a term 2 things are required which are entity or concept set and another is an isA relation between terms. If an isA relation is maintained between terms then hypernym term is a concept term. And the hyponym terms is a entity term. If no isA relation is maintained between terms then its type can be decided individually. For example concept of a terms "Apple and Microsoft" is company.

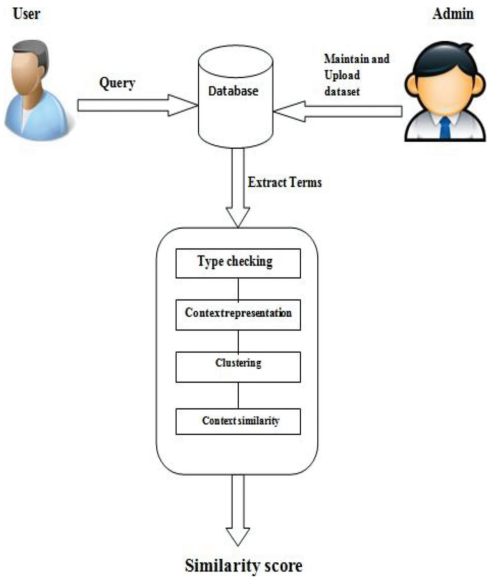

Figure 1. Architecture of semantic similarity.

Module 3: Context Representation

A context of a given term can be extracted from its type. A context of term is depending on its type so, that a type of a term can be input to it. A context can be an entity, if a term is concept. And if a given term is an entity then, its context can be concept. For example concept contexts of the term Apple are fruit, company, food, seasonal fruit, and tree.

Module 4: Concept Clustering

Concept clustering algorithm is added into Refined approach algorithm as a part of it. For finding similarity clustering algorithm was implemented as a part of refined approach algorithm. To identify multiple senses of terms K – medoid clustering algorithm is used. A clustering algorithm takes a collection of concept as an input. By using this clustering algorithm similar context or senses of a term are grouped together. For example fruit, seasonal fruit and tree fruit are grouped together into one cluster because all contexts of term Apple are having same sense.

Module 5: Context Similarity

To estimate similarity between two contexts a similarity function F (.) can be used. The similarity can be measured as Sim(Tt1, Tt2) = F(Tt1, Tt2). A similarity function F (.) can be any one of the evaluation function such as, cosine and Jaccard. Another methods used for finding similarity are Max: sim(Tt1; Tt2), Average: sim(Tt1; Tt2), Weighted: sim(Tt1; Tt2).

5 Experimental Result

Module 1

Figure 2 Shows the dataset which consists of collection of word pairs. Each word pair has assigned id and type.wordsim353 dataset is used.

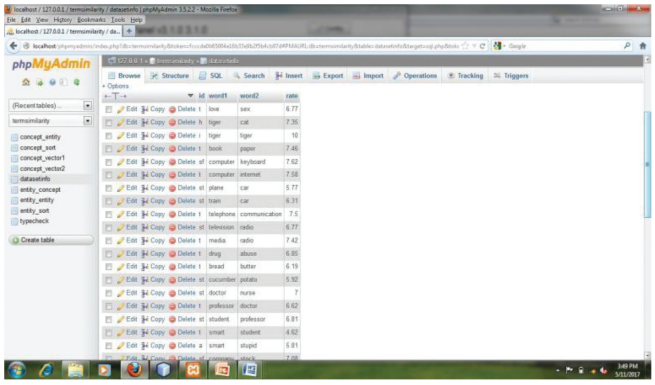

Figure 2. Candidate set of words.

Module 2: Type Checking

Following Figure 3 shows result of type checking in which two word pairs are given as input having isA relation between words.

Module 3: Context Representation

Following three figures shows the result of context representation in which two terms are given as input to it. Context of word pair can be determined according to its id assigned in a dataset.

Module 4: Concept Clustering

Following figure shows the result of clustering in which two terms are given as an input to it. Clusters of word pair can be generated according to its id and family from which it belongs.

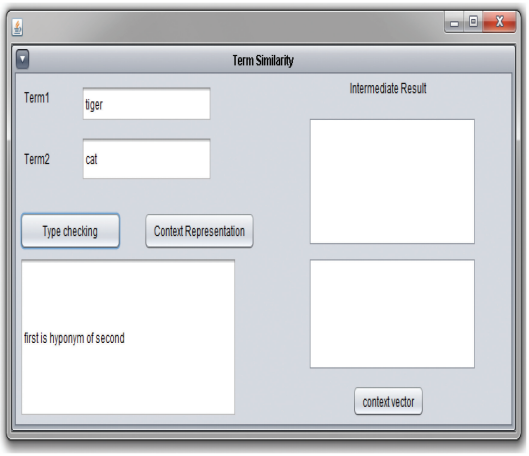

Figure 3. Type checking.

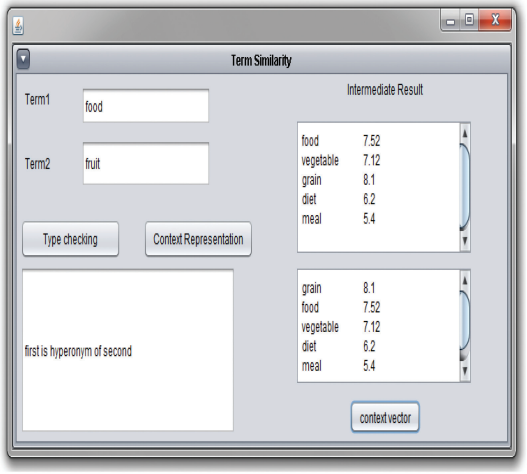

Figure 4. Context representation (id = hr).

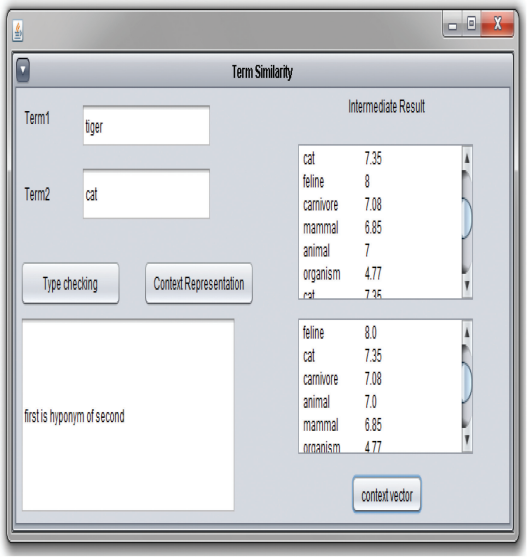

Figure 5. Context representation (id = h).

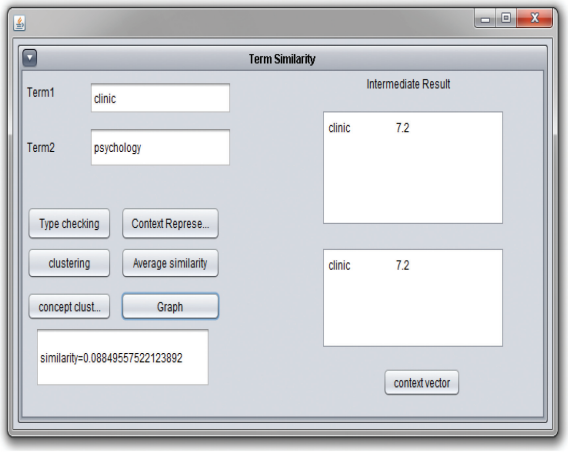

Figure 6. Concept clustering.

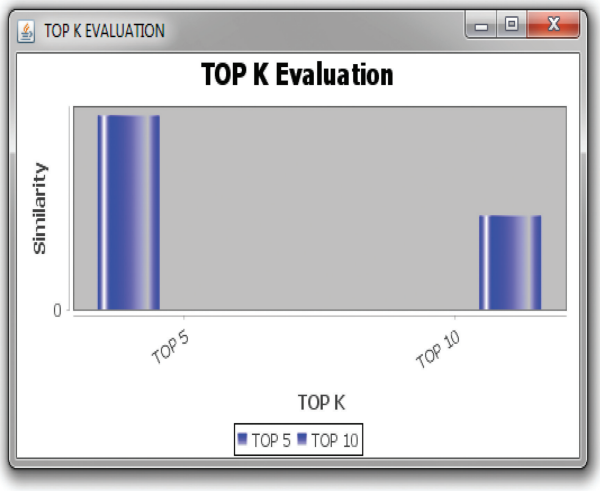

Chart 1. Similarity of words.

Module 5: Context Similarity

Following figure shows the result of similarity between two words in terms of graphical representation.

6 Conclusions

This approach is an efficient and effective for computing semantic similarity between terms. isA semantic network is present between pair of words. Concept clustering algorithm is introduced to avoid ambiguous terms. Finally max similarity function is used to compute similarity between two terms. This method is efficient enough to applied on large scale dataset. The future work of this system is how to apply same technique on short text categorization.

References

[1] A. Budanitsky and G. Hirst, "Evaluating wordnet-based measures of lexical semantic relatedness," Comput. Linguistics, vol. 32, pp. 13–47, 2006.

[2] M. G€artner, A. Rauber, and H. Berger, "Bridging structured and unstructured data via hybrid semantic search and interactive ontology-enhanced query formulation," Knowl. Inf. Syst., vol. 41, pp. 761–792, 2014.

[3] G. A. Miller, "WordNet: A lexical database for english," Commun. ACM, vol. 38, no. 11, pp. 39–41, 1995.

[4] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in Proc. 14th Int. Joint Conf. Artif. Intell., 1995, pp. 448–453.

[5] A new semantic relatedness measurement using wordnet features, M. A. H. Taieb, M. B. Aouicha, and A. B. Hamadou, 2013.

[6] Y. Wang, H. Li, H. Wang, and K. Q. Zhu, "Concept-based web-search," in Proc. 31st Int. Conf. Conceptual Model. ER, 2012, pp. 449–462.

[7] R. Rada, H. Mili, E. Bichnell, and M. Blettner, "Development and application of a metric on semanticnets," IEEE Trans. Syst., Man Cybern., vol. 9, no. 1, pp. 17–30, Jan./Feb. 1989 E. Agirre, M. Cuadros, G. Rigau, and A. Soroa, "Exploring knowledge bases for similarity," in Proc. 7th Int. Conf. Language ResourcesEval., 2010, pp. 373–377.

[8] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in Proc. 14th Conf. Compute. Linguistics, 1992, pp. 539–545.

[9] Y. Song, H. Wang, Z. Wang, H. Li, and W. Chen, "Short text conceptualization using a probabilistic knowledgebase," in Proc. 22nd Int. Joint Conf. Artif. Intell., 2011, pp. 2330–2336.

[10] W. Wu, H. Li, H. Wang, and K. Q. Zhu, "Probase: A probabilistic taxonomy for text understanding," in Proc. ACM SIGMOD Int. Conf. Manag. Data, 2012, pp. 481–492.

[11] Ryan Shaw, Member, Anindya Datta, "Building a Scalable Database-Driven Reverse Dictionary", 2013.

[12] Wenhao Wang, BenFu Tang, Cheng Zhu, Bin Liu, Aiping Li, Zhaoyun Din, "Clustering Using a Similarity Measure Approach Based on Semantic Analysis of Adversary Behaviors", IEEE 2020.