A review of the end-to-end semantic disambiguation literature

Xia Linhan
Beijing normal university-Beijing normal university-Hong Kong Baptist university united international college
Department of Computer science and technology
Zhu Hai, China
temp.xialinhan@gmail.com

***Abstract*—This review provides a comprehensive look at the literature related to WordNet-based end-to-end semantic understanding. In recent years, end-to-end semantic understanding has made significant progress in the field of natural language processing (NLP), providing artificial intelligence systems with the ability to understand and generate human language. In particular, semantic understanding has shown great value and potential in applications such as information retrieval, sentiment analysis, machine translation, and intelligent assistants. And WordNet, as a large lexical database, provides a powerful tool for understanding the complex semantic relationships between words.This review first outlines the basic concepts and techniques of semantic understanding, including semantic analysis, semantic understanding models, and recent developments. Next, we discuss in detail the application of WordNet to semantic understanding and how its semantic networks and word-sense relations can be used to enhance semantic understanding. We also review a number of important studies on end-to-end semantic understanding systems, particularly those based on deep learning and neural network models.The analysis of this literature reveals the important role of WordNet-based end-to-end semantic understanding in improving the explanatory power of models, enhancing system performance, and addressing issues such as semantic ambiguity in practical applications. However, a number of challenges remain in the current research, including how to integrate and use WordNet's resources more effectively, how to handle multilingual and domain-specific semantic understanding, and how to design more robust and adaptable semantic understanding models.**

***Index Terms*—NLP, WordNet, Semantic analysis**

I. Background

As we move into the new era of Web 3.0, technological advances in artificial intelligence and machine learning are already having a profound impact on the way information is accessed and processed. the idea of Web 3.0 relies on the ability of machines to understand and process web content, which is what the field of natural language processing (NLP) - and in particular end-to-end semantic understanding - can provide. This is what the field of Natural Language Processing (NLP) - particularly end-to-end semantic understanding - can provide.

Web 3.0, also known as the Semantic Web, stands for the next generation of the Internet, whose main goal is the semanticisation of data and information. Under this concept, information and services will be precisely categorised and targeted so that machines can better understand the needs of users and provide more personalised services. In such a context, the importance of end-to-end semantic understanding technology is clear. It covers all aspects of natural language processing (NLP), including word disambiguation, semantic role annotation of sentences, semantic dependency analysis, knowledge graphs, etc., which are fundamental to the understanding, generation and use of human language.

End-to-end semantic understanding aims to obtain deep semantic information from the input text, not just surface-level lexical or syntactic information. For example, it can help systems understand the intent of a user's query and respond more accurately to the user's needs. Similarly, by understanding content on social media, it can help companies better understand the public's emotional inclination towards their brand or product.

However, there are a number of challenges to achieving end-to-end semantic understanding. Firstly, the complexity and diversity of language makes it difficult to understand and parse natural language text. In addition, the ambiguity of words is another major challenge. The same word may have different meanings in different contexts, and understanding these meanings correctly is crucial for semantic understanding. Secondly, processing large amounts of unstructured data is also a challenge, as it requires significant computational resources and efficient algorithms.

One effective way to address these challenges is a semantic understanding approach based on WordNet, a large lexical database that contains a large number of words and their semantic relationships. These relationships can help us understand complex relationships between words, deal with word sense ambiguity, and understand complex language structures and contexts. In addition, WordNet provides us with a convenient tool for representing and querying lexical knowledge, which is essential for building complex semantic understanding systems.

Overall, WordNet-based end-to-end semantic understanding is an important foundational technology for the Web 3.0 era, providing us with a smarter, more automated way to understand and process information on the Web. However, it requires more in-depth research and development to address the various challenges of natural language processing, including dealing with the complexity and diversity of languages,

handling large amounts of unstructured data, and resolving word sense ambiguities. In addition, we need to explore how we can better integrate and utilise resources such as WordNet to improve our semantic understanding and enable us to better adapt to the needs of the Web 3.0 era.

II. Review of WordNet

Wordnet, a large English dictionary built and maintained by the Laboratory for Cognitive Science at Princeton University under the direction of George A. Miller, Professor of Psychology,1 is one of the key tools for natural language processing and computational linguistics. the main feature of Wordnet is that English words are grouped into sets of synonyms (synsets) according to their lexical meanings, and these synsets are connected by different types of semantic relations and lexical relations connect these synsets to form a complex semantic network2.

Wordnet contains four lexical synsets: noun, verb, adjective and adverb, each of which has a short definition (gloss) and some example sentences to illustrate its usage. the most common semantic relation in Wordnet is the hypernymy and hyponymy relation, which indicates that one concept is a For example, fruit is a superlative of apple and apple is a sublative of fruit. Such relations can form a hierarchy, from the most abstract ENTITY to the most concrete instance.3 In addition to subordination relations, Wordnet also contains other types of relations, such as whole-part relations (meronymy and holonymy), antonymy relations (antonymy), derivationally related form) and so on2.

As a large-scale knowledge base of English, Wordnet has a wide range of applications, such as text understanding, text generation, information retrieval, machine translation, question and answer systems, etc. Wordnet can provide information on semantic similarity, semantic compatibility and semantic implication between words, thus helping computers to deal with ambiguities, multiple meanings and metaphors in natural language.4 Wordnet can also be used as an aid to help people learn English, find synonyms or antonyms, understand relationships between words, etc.

Although Wordnet has been in development for decades, there are still many areas for improvement and exploration, such as how to expand and update the data in Wordnet, how to improve the quality and consistency of the data in Wordnet, how to use other resources and technologies to enrich and optimise Wordnet, how to integrate Wordnet with knowledge bases from other languages or domains, and how to evaluate and validate the effectiveness of Wordnet for different tasks, etc.

III. Calculate similarity through WordNet

Calculating text Similarity The most important part of our project is to evaluate the ambiguity introduced in the end-to-end transmission by comparing the semantics understood on both sides of the communication, and finally to optimize our results with an optimization algorithm for disambiguation.

Calculating text similarity is an important task in natural language processing, which refers to the evaluation of the degree of similarity or relatedness of two or more texts at the semantic level. There are a wide range of applications for computing text similarity, such as information retrieval, question and answer systems, machine translation, text summarisation, etc. Methods for computing text similarity can be classified into corpus-based methods, knowledge-base based methods and deep learning based methods1.

One is based on the bag-of-words or vector space model, which represents text as vectors of features such as word frequency or TF-IDF, and then calculates the cosine similarity or Euclidean distance between the vectors The other is based on a topic model, which represents the text as a vector of potential topic distributions and then calculates the KL scatter or JS scatter between the vectors.3 The advantage of the corpus-based approach is that it is simple to implement and does not rely on external knowledge, but the disadvantage is that it ignores the semantic relationships between words and the structural information between texts, and is vulnerable to problems such as data sparsity and polysemous words. problems.

There are two main ideas: one is based on dictionaries or encyclopaedic knowledge bases, such as WordNet and Wikipedia, which use the synonymy, antonymy, subordination and whole-part relationships between words defined in them to calculate the similarity between texts; the other is based on ontology knowledge bases The other is based on ontology-like knowledge bases, such as DBpedia, YAGO, etc., which use the attribute relations, type relations, instance relations, etc. between the entities described therein to calculate the similarity between texts. The advantage of the knowledge-base based approach is that it can use the rich semantic information to improve the accuracy of similarity calculation, but the disadvantage is that it depends on the quality and coverage of the knowledge base, as well as the alignment and mapping between the knowledge base and the text.

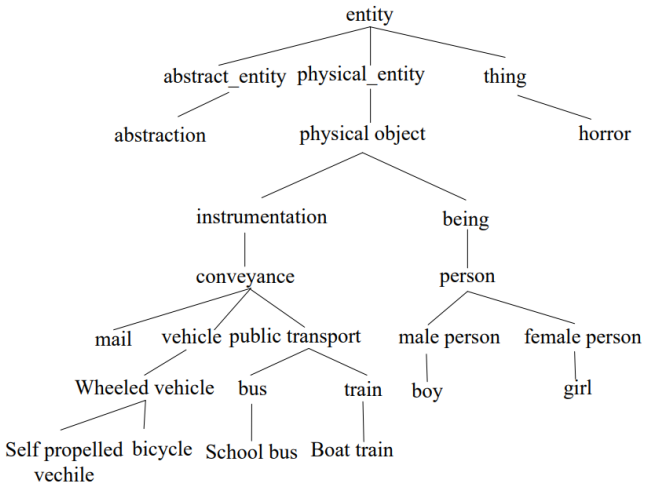

Fig. 1. A Fragment of is-a Relation in WordNet

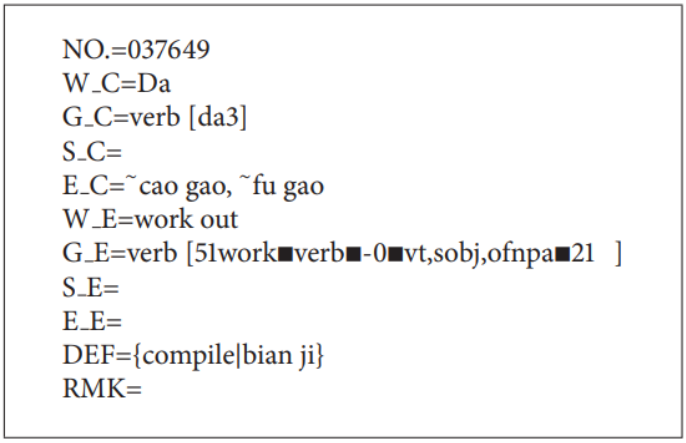

Fig. 2. Example of Chinese Word concept

Deep learning-based approaches use neural network models to learn the similarity between texts, and there are two main ideas: one is based on an encoder-decoder structure, where two texts are encoded separately as vector representations and a decoder is used to predict their similarity scores; the other is based on an attention mechanism, which encodes two texts as a sequence of vectors and then captures the alignment and matching relationship between them through an attention mechanism, and then outputs the similarity score through an aggregation layer. The advantage of deep learning-based approaches is that they can automatically learn complex and non-linear similarity functions between texts, taking into account the semantic and structural information of words and texts, but the disadvantage is that they require large amounts of annotated data and computational resources, as well as a lack of interpretability and generalisation.

Semantic similarity measurement is a core problem in artificial intelligence, psychology and cognitive science. It is widely used in natural language processing, information retrieval, word sense disambiguation, text segmentation, question answering, recommendation systems, information extraction and other fields. Wordnet-based measures have attracted a lot of attention in recent years. They demonstrated their talents to make these applications more intelligent. Many semantic similarity measures have been proposed. Overall, all metrics can be classified into four categories: path length based metrics, information content based metrics, feature based metrics, and hybrid metrics. The characteristics, performance, advantages and disadvantages of different measures are discussed, and suggestions for future research are proposed.

To achieve higher semantic accuracy, we use WordNet for similarity calculation and evaluation model construction.

Before starting our review, we need to first define multiple related concepts, which are necessary in our discussion.

1) $len(c_i, c_j)$:the length of the shortest path from synset c_i to synset c_j in WordNet.
2) $lso(c_i, c_j)$:the lowest common subsumer of c_i and c_j.
3) $depth(c_i)$:the length of the path to synset c_i from the global root entity, and $depth(root) = 1$.
4) $deep_{max}$:the max depth(c_i) of the taxonomy.
5) $hypo(c)$:the number of hyponyms for a given concept c.
6) $node_{max}$:the maximum number of concepts that exist in the taxonomy.
7) $sim(c_i, c_j)$:semantic similarity between concept c_i and concept c_j.

Now let's start an overview of the different algorithms we found:

A. Path-based Measures

Path-based measures operate on the principle that the relationship between two concepts is determined by a combination of the path's characteristics, including its length and the respective positions of the concepts within the taxonomy.

B. The Shortest Path based Measure

The measure exclusively focuses on the distance between c_1 and c_2, disregarding other factors. It posits that the similarity (c_1, c_2) relies on the relative closeness of the two concepts within the taxonomy. In essence, this measure represents a modified version of the distance method. It is derived from two fundamental observations. Firstly, the behavior of conceptual distance displays similarities to metric properties. Secondly, the conceptual distance between two nodes is directly correlated with the number of hierarchical edges separating them.

$$sim_{path}(c_1, c_2) = 2 \times deep_{max} - len(c_1, c_2) \quad (1)$$

According to equation (1) it is need to recognized that,

1) On the specific version of WordNet, $deep_{max}$is a constant value. The similarity of two concept is $len(c_1, c_2)$from c_1 to c_2.
2) If $len(c_1, c_2)$equal to zero, the similarity between two concepts is the max value: $depth_{max}$. If the $len(c_1, c_2)$is the max value, the similarity is zero, according the situation, we can get the similarity between zero and $depth_{max}$.

C. Wu & Palmer's Measure

Wu and Palmer have proposed a sophisticated similarity metric. This innovative approach contemplates the relative positions of concepts c_1 and c_2 within the taxonomy, considering the location of their most specific shared concept, defined as $lso(c_1, c_2)$. The assumption guiding this model suggests that the degree of similarity between any two concepts is dependent upon both the length of their connecting path and the depth within the path-based measurement system. This new angle on similarity metrics encourages a deeper understanding of the intricate relationships between concepts in a given taxonomy.

$$sim_{WP}(c_1, c_2) = \frac{2 \times depth(lso(c_1, c_2))}{len(c_1, c_2) + 2 \times depth(lso(c_1, c_2))} \quad (2)$$

According to (2),it is need to recognized that:

1) The similarity between two concepts (c_1, c_2) is the function of their distance and the lowest common subsumer($lso(c_1, c_2)$.
2) If the $lso(c_1, c_2)$ is root, $depth(lso(c_1, c_2)) = 1$,$sim_{WP}(c_1, c_2) > 0$; if the two concepts have the same sense, the concept c_1, concept c_2 and $lso(c_1, c_2)$ are the same node. $len(c_1, c_2) = 0$. $simWP(c_1, c_2) = 1$; otherwise $0 < depth(lso(c_1, c_2)) < deep_max$, $0 < len(c_1, c_2) < 2 \times deep_max$, $0 < sim_{WP}(c_1, c_2) < 1$. Thus, the values of $sim_{WP}(c_1, c_2)$ are in (0, 1].

D. Information Content-based Measure

In WordNet, we can assume that each concept contains a wealth of information. To assess the similarity between concepts, we rely on the information content of the individual concepts. In fact, the similarity measure between concepts depends on how much general information they share. When two concepts share more common information, we can consider them to be more similar. This similarity measure provides us with a more accurate and precise basis for associativity assessment.

E. Resnik's Measure

In 1995, Resnik proposed a similarity measure based on information content. The method assumes that for two given concepts, the similarity depends on the information content that includes them in the classification system.

Information content is a measure of the information covered by a concept in a taxonomy. It can be used to measure the relevance and similarity between concepts. Resnik's method assesses the degree of similarity between these two concepts by taking their information content as a metric. Specifically, two concepts are more similar to each other if they are covered by a higher level concept in the classification system.

This similarity measure based on information content provides a more accurate and precise way to compare similarities between concepts. By considering the position and hierarchy of concepts in the classification system, this method can better capture the semantic correlation between concepts. Therefore, more reliable and accurate concept similarity evaluation results can be obtained using this method, which can provide more useful information for tasks in fields such as natural language processing and knowledge representation.

$$sim_{Resnik}(c_1, c_2) = -logp(lso(c_1, c_2)) = IC(lso(c_1, c_2)) \tag{3}$$

According to the (3)we have to notice:

1) The values only rely on concept pairs' lowest subsumer in the taxonomy.
2) $lso(mail, vehicle) = lso(mail, bicycle) = conveyance$, therefore $sim_{Resnik}(mail, vehicle) = sim_{Resnik}(mail, bicycle) = IC(conveyance)$

F. Jiang's Measure

Jiang obtained semantic similarity by calculating semantic distance, which is the opposite of distance.

$$dis_{Jiang}(c_1, c_2) = (IC(c_1) + IC(c_2) - 2IC(lso(c_1, c_2) \tag{4}$$

According to (4),we have to recognize:

1) This measure takes into account the mutual influence between the compared concepts separately.
2) Its measurement is the semantic distance between two concepts. Semantic similarity is the opposite of semantic distance.

Within the realm of information content-based similarity measurements, all of the methods presented in previous strive to leverage the inherent information in an optimal way to assess the similarity amongst concept pairs. Consequently, the method employed to derive IC is of paramount importance as it directly impacts the overall performance. Typically, five methods are in use. The first method involves deriving IC through a meticulous statistical analysis of corpora , from which the likelihood of concept occurrences can be deduced. This method postulates that if we denote the probability of encountering an instance of a concept c in the taxonomy as *p(c)*, then *IC(c)* can be represented as the negative logarithm, *log p(c)*. This implies that as the probability escalates, *IC* correspondingly diminishes.

$$IC(c) = -logp(c) \tag{5}$$

Probability of a concept was estimated as:

$$p(c) = \frac{freq(c)}{N} \tag{6}$$

Where N represents the total number of nouns and $freq(c)$ stands for the frequency of an instance of concept c in the taxonomy. When calculating $freq(c)$, any noun or its taxonomical hyponyms that occur in the given corpora are included. This means if $c1$ is a subtype of $c2$, then $p(c1) < p(c2)$. Therefore, the more abstract a concept is, the higher its associated probability and the lower its information content.

$$Freq(c) = \sum_{w \in W(c)} count(w) \tag{7}$$

The method Nuno proposed makes use of hyponyms to calculate IC and leverages WordNet as a statistical resource. The IC value of a concept is seen as a function of the number of its hyponyms. For each concept, the abstraction level increases with the number of its hyponyms. Therefore, concepts with more hyponyms convey less information compared to leaf concepts. In the hierarchy of the taxonomy, the root node is the least informative, while leaf nodes are the most informative. The IC of the root node is $IC(root) = 0$, and for leaf nodes it's $IC(leaf) = 1$. As we navigate from leaf nodes towards the root node, the IC value monotonically decreases, ranging from 1 to 0. This method is simple and independent of corpus. However, concepts with the same number of hyponyms have

the same IC values, and all leaf nodes will also have the same IC values, even though they are at different levels in the taxonomy. For example, $IC(mail) = IC(bicycle)$.

$$IC(c) = 1 - \frac{log(hypo(c) + 1)}{log(node_{max})} \tag{8}$$

The third is based on the assumption that the classification leaves represent the semantics of the most specific concepts in the domain in WordNet, and the more leaves a concept has, the less information it expresses.

$$IC(c) = -log(\frac{\frac{|leaves(c)|}{|subsumers(c)|} + 1}{max_{leaves} + 1}) \tag{9}$$

Given C as the set of all concepts in the ontology and c as a particular concept, we define $leaves(c) = l|l \in hyponyms(c) \wedge l$ is a leaf. $Subsumers(c) = a \in C|c \leq a \cup c$, where$c \leq a$ signifies that c is a hierarchical specialization of a. Max_leaves denotes the maximum number of leaves pertaining to the root node of the hierarchy. This method doesn't consider the depth of leaves, hence concepts with the same number of leaves will possess the same IC values. For example, $IC(vehicle) = IC(wheeled, vehicle)$.The fourth method presumes that each concept is sufficiently defined with ample semantic embedding within the organizational structure, property functions, property restrictions, and other logical assertions. Here, the IC value is a function of both relations and hyponyms. A weight factor is employed to adjust each part's contribution.

$$IC(c) = \rho \cdot IC_{rel}(c) + (1 - \rho)_{Nuno}(c) \tag{10}$$

$$IC(c) = \frac{log(rel(c) + 1)}{log(rel_{max} + 1)} \tag{11}$$

$$\rho = \frac{total_{rel} + 1)}{log(rel_{max}) + log(node_{max})} \tag{12}$$

Where $rel(c)$ denotes the number of relations of concept c and rel_{max} denotes the total number of relations. The last algorithm assumes that each concept is unique in the classification and the IC value is a function of the concept topology, which can effectively distinguish different concepts and obtain a more accurate IC value. It is defined as:

$$IC(c) = \frac{log(depth(c))}{log(deep_{max})} \times (1 - \frac{log(\sum_{a \in hypo(c)} \frac{1}{depth(a)} + 1)}{log(node_{max})}) \tag{13}$$

Where for a given concept c, a is a concept of the taxonomy such that $a \in hypo(c)$. If c is the root, then $depth(root)$ is 1 and $log(deep(c))$ is 0. If c is a leaf node, $hypo(c)$ is 0. And then,

$$\sum_{a \in hypo(c)} \frac{1}{depth(a)} = 0 \tag{14}$$

What's more:

$$IC(c) = \frac{log(depth(c))}{log(deep_{max})} \tag{15}$$

Because sparse data problem is not avoided in Corpora-dependent IC Metric, corporaindependent IC Metric is popular.

G. Feature-based Measure

In contrast to the measures discussed previously, the feature-based measure is not dependent on the taxonomy or the concept subsumers. It strives to extract similarity values by leveraging the inherent properties of the ontology. It operates under the assumption that each concept is illustrated by a set of words that demonstrate its unique features or properties, such as definitions or WordNet "glosses". The degree of similarity between two concepts is gauged by the quantity of shared attributes and the scarcity of unique attributes. The more shared traits two concepts have, and the fewer unique traits they possess, the greater the similarity between them.

Tversky's model is a classic example of this approach. It posits that similarity is not necessarily symmetric. Features shared between a subclass and its superclass contribute more to the evaluation of similarity than those shared in the opposite direction. This introduces a level of asymmetry where the similarity of a subclass to its superclass is not identical to the similarity of the superclass to its subclass. Tversky's model is defined as :

$$sim_{Tversky}(c_1, c_2) = \frac{|C_1 \cap C_2|}{|C_1 \cap C_2| + k|C_1/C_2| + (k-1)|C_1/C_2|} \tag{16}$$

Where C_1, C_2 correspond to description sets of concept c_1 and c_2 respectively, k is adjustable and $k \in [0, 1]$.

IV. Sentence Semantic Similarity Calculation based on WordNet

WordNet, initially developed at Princeton University, has become a staple in the field of computational linguistics due to its extensive lexical database of English. It is not just a simple thesaurus; instead, it maps out the relationships between words, effectively serving as a semantic network where words are interconnected through their semantic relationships, such as synonyms, antonyms, hypernyms, and hyponyms. This design has made WordNet a powerful tool for tasks involving semantic understanding, like measuring the semantic similarity between sentences.

Researchers have proposed several methods to calculate sentence similarity based on WordNet. These techniques typically involve computing the similarity between individual words using WordNet and then aggregating these word-level similarities to derive sentence-level similarity. These methods vary in how they measure word similarity, how they align words between sentences, and how they aggregate word-level similarity scores. For instance, one approach uses path-based measures that capture the shortest path between two words within WordNet's hierarchy, while others may incorporate information content of words or concepts.

Jiang and Conrath's method is a classic example, measuring semantic distance by incorporating both the path length and the information content of the Least Common Subsumer (LCS). Resnik's method is another popular approach that measures word similarity solely based on the information content of the LCS. Recently, more sophisticated techniques have also been proposed, such as those using machine learning or neural network models to learn word representations from WordNet's structure and use these learned representations to calculate word and sentence similarity.

Here are the various algorithms we studied and introduced in detail:

A. Algorithm Introduction of Sentence Similarity

In the realm of computational linguistics, sentence similarity algorithms have been largely classified into three distinctive categories: keyword-based methods, semantic-based methods, and syntactic structure-based methods. These methods, each with their unique characteristics, tackle the issue of sentence similarity from different perspectives.

The keyword-based strategies primarily employ algorithms such as string matching, leveraging the superficial lexical components of sentences. Conversely, the semantic-based strategies employ methods such as the TF-IDF, utilizing a vector space model that emphasizes the significance of semantics in information retrieval. There are also methods that calculate sentence similarity based on semantic dictionaries or dependency analysis, which focus on the semantic relationships between words in a sentence.

These proposed methodologies have significantly contributed to advancing research in the field of Chinese sentence similarity. However, they are not without their drawbacks. For instance, the keyword-based approach focuses heavily on superficial lexical attributes of the words, which often results in a lack of deep semantic understanding. The use of semantic dictionaries compensates for this lack of depth to some extent, but its effectiveness is limited by the constraints of the corpus and the handling of unknown words. On the other hand, dependency analysis methods consider the sentence's syntactic structure but often entail complex algorithms and high computational costs.

To address these challenges, this paper introduces a new approach for calculating sentence similarity, which combines syntactic structure information, keyword semantic information, and other factors. This method, grounded on syntactic structure, aims to enhance the accuracy of sentence similarity computations by incorporating a more comprehensive understanding of sentence semantics.

B. Xiao Li and Qingsheng Li's work

The methodology involves dividing the sentence into three parts: the subject, predicate, and object components. This division is facilitated through a process called 'Language Technology Platform (LTP) analysis', which tags parts of the sentence according to the "HED" scheme. According to the paper, the subject component is located on the left of the predicate component, while the object component is on its right.

Crucially, the paper stresses the importance of both key components (subject, predicate, object) and modifier components (attributive, adverbial, and complement) in a sentence. It proposes that while key components are usually given more weight in sentence similarity calculations, modifier components can also significantly impact the computed similarity.

To illustrate this point, the paper provides an example of two sentences, "I like her red and pink face" and "I like his naughty and lovely face". Using only key components, these sentences would be deemed entirely similar. However, taking modifier components into account allows for a more nuanced understanding of sentence similarity.

The paper also discusses a few issues encountered while dealing with modifiers. For instance, it mentions the challenge of calculating some modifiers such as adverbs. To address these issues, it suggests considering all nouns, verbs, pronouns, and adjectives in a sentence while calculating sentence similarity. Additionally, it also considers numerals modifying different quantifiers and adverbs stating "no" in negative sentences.

The paper concludes this section by offering weighted values to different elements of a sentence - nouns, verbs, adjectives, and pronouns are assigned weights of 0.3, 0.3, 0.2, and 0.2 respectively, reflecting their contribution to the sentence's meaning.

Overall, the main objective of the paper seems to be the enhancement of sentence similarity computation accuracy through a more comprehensive consideration of different sentence components and their respective contributions to sentence meaning.

Considering various aspects, this paper puts forward the following calculation formula of sentence semantic similarity:

$$Sim(S_1, S_2) = \beta_1 \cdot \beta_2 \tag{17}$$

$$\beta_1 = l \cdot \lambda \cdot \gamma \cdot \phi \tag{18}$$

$$\beta_2 = [\alpha_1 Sim_1(B_1, B_2) + \alpha_2 | + Sim + 3(B_1, B_2) | + \alpha_3 (B_1, B_2)] \tag{19}$$

1) The parameter k, termed the sentence pattern adjustment coefficient, is designed to calibrate the computation of sentence similarity according to varying sentence patterns. Given that interrogative sentences bear a significantly different tone compared to other sentence types, k is set to 0.1 when comparing an interrogative sentence to imperative, declarative, or exclamatory sentences. When evaluating similarity between other sentence patterns, the coefficient is set to 0.5. When sentences share the same pattern, k equals 1.
2) The parameter λ, also known as the sentence component coefficient, acts as an adjustment factor when sentences are composed of an unequal number of components. Its value is defined as $2i/(m+n)$, where m and n denote the count of components in sentences S_1 and S_2,

respectively, and i represents the quantity of matching components within S_1 and S_2.

3) The parameter γ, identified as the negativity coefficient, comes into play when the predicate heads in S_1 and S_2 are overtly antonymous or if "no" precedes the predicate head in S_1 relative to S_2. In such circumstances, the value of γ is set to -1.
4) The parameter ϕ serves as an auxiliary coefficient, activated when the predicate heads of two sentences are antonyms and the subject and object in both sentences are switched. Under such conditions, the value of ϕ is set to -1.
5) Post syntactic analysis, a sentence is segmented into three components, leading to the determination of β_2 based on Sim1(S_1, S_2), Sim2(S_1, S_2), and Sim3(S_1, S_2), which denote the similarity of the subject component, predicate component, and object component, respectively. Due to the potential negativity of Sim2(S_1, S_2), its absolute value is considered. α_1, α_2, and α_3 are assigned as coefficients for the three respective parts, with weights of 0.3, 0.5, and 0.2, guided by the relative contribution of the sentence components and practical observations.

Moreover, when aligning components of the sentences for comparison, situations might arise wherein modifiers have varying structures or words exhibit differing quantities . To streamline the calculation process, the following stipulations are established for comparing identical components. When the matching components of the two sentences.

REFERENCES

[1] Y. Li, Z. A. Bandar, and D. McLean, An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources," IEEE Transactions on Knowledge and Data Engineering, vol. 15, Issue 4, pp. 871 – 882, July-August 2003.

[2] R. K. Srihari, Z. F. Zhang and A. B. Rao, Intelligent indexing and semantic retrieval of multimodal documents," Information Retrieval, vol. 2, pp. 245-275, 2000.

[3] R. Rada, H. Mili, E. Bicknell and M. Blettner, Development and Application of a Metric on Semantic Nets," IEEE Transactions on Systems, Man and Cybernetics, vol. 19, Issue 1, pp. 17 - 30, January-February 1989.

[4] S. Patwardhan, S. Banerjee and T. Pedersen, Using measures of semantic relatedness for word sense disambiguation," Proceedings of 4th International Conference on Computational Linguistics and Intelligent Text Processing and Computational Linguistics, February 16-22 2003, Mexico City, Mexico.

[5] H. Kozima, Computing Lexical Cohesion as a Tool for Text Analysis," doctoral thesis, Computer Science and Information Math., Graduate School of Electro-Comm., Univ. of Electro- Comm., 1994.

[6] A. G. Tapeh and M. Rahgozar, A knowledge-based question answering system for B2C eCommerce," Knowl.-Based Syst., vol. 21, no. 8, 2008.

[7] Y. Blanco-Fernández, J. J. Pazos-Arias, A. Gil-Solla, M. Ramos-Cabrer, M. López- Nores, J. García-Duque, A. Fernández-Vilas, R. P. Díaz-Redondo and J. Bermejo-Muñoz, A flexible semantic inference methodology to reason about user preferences in knowledge-based recommender systems," Knowl.-Based Syst., vol. 21, no. 4, 2008.

[8] J. Atkinson, A. Ferreira and E. Aravena, Discovering implicit intention-level knowledge from naturallanguage texts," Knowl.-Based Syst., vol. 22, no. 7, 2009.

[9] M. Stevenson and M. A. Greenwood, A semantic approach to IE pattern induction," Proceedings of 43rd Annual Meeting on Association for Computational Linguistics, June 25-30 2005, Ann Arbor, Michigan, USA.

[10] C. Fellbaum, ed., WordNet: An electronic lexical database," Language, Speech, and Communication. MIT Press, Cambridge, USA, 1998.

[11] H. Bulskov, R. Knappe and T. Andreasen, On Measuring Similarity for Conceptual Querying," Proceedings of the 5th International Conference on Flexible Query Answering Systems, October 27-29 2002, Copenhagen, Denmark.

[12] Z. Wu and M. Palmer, Verb semantics and lexical selection," Proceedings of 32nd annual Meeting of the Association for Computational Linguistics, June 27-30 1994, Las Cruces, New Mexico.

[13] C. Leacock and M. Chodorow, Combining Local Context and WordNet Similarity for Word Sense Identification, WordNet: An Electronic Lexical Database," MIT Press, 1998, pp. 265-283.

[14] P. Resnik, Using information content to evaluate semantic similarity," Proceedings of the 14th International Joint Conference on Artificial Intelligence, August 20-25 1995, Montréal Québec, Canada.

[15] D. Lin, An information-theoretic definition of similarity," Proceedings of the 15th International Conference on Machine Learning, July 24-27 1998, Madison, Wisconsin, USA.

[16] J. J. Jiang and D. W. Conrath, Semantic similarity based on corpus statistics and lexical taxonomy," Proceedings of International Conference on Research in Computational Linguistics, August 22-24 1997, Taipei, Taiwan.

[17] A. Tversky, Features of Similarity," Psycological Review, vol. 84, no. 4, 1977.

[18] M. A. Rodriguez and M. J. Egenhofer, Determining Semantic Similarity among Entity Classes from Different Ontologies," IEEE Trans. on Knowledge and Data Engineering, vol. 15, no. 2, 2003.

[19] H. Dong, F. K. Hussain and E. Chang, A Hybrid Concept Similarity Measure Model for Ontology Environment," Lecture Notes in Computer Science on the Move to Meaningful Internet Systems: OTM 2009 Workshops, vol. 5872, 2009.

[20] Z. Zhou, Y. Wang and J. Gu, New Model of Semantic Similarity Measuring in Wordnet," Proceedings of the 3rd International Conference on Intelligent System and Knowledge Engineering, November 17-19 2008, Xiamen, China.

[21] N. Seco, T. Veale and J. Hayes, An intrinsic information content metric for semantic similarity in WordNet," Proceedings of the 16th European Conference on Artificial Intelligence, August 22-27 2004, Valencia, Spain.

[22] D. Sánchez, M. Batet and D. Isern, Ontology-based information content computation," Knowl.-Based Syst., vol. 24, no. 2, 2011.

[23] Md. H. Seddiqui and M. Aono, Metric of intrinsic information content for measuring semantic similarity in an ontology," Proceedings of 7th Asia-Pacific Conference on Conceptual Modeling, January 18-21 2010, Brisbane, Australia.

[24] H. Rubenstein and J. B. Goodenough, Contextual correlates of synonymy," Communications of the ACM, vol. 8, no. 10, 1965.