

Article

Integrated Subject–Action–Object and Bayesian Models of Intelligent Word Semantic Similarity Measures

Siping Zeng ¹, Xiaodong Liu ² , Wenguang Lin ³ , Vasantha Gokula ²  and Renbin Xiao ^{4,*} 

¹ School of Economics and Management, Xiamen University of Technology, Xiamen 361024, China; 2015000023@xmut.edu.cn

² School of Computing, Engineering, and the Built Environment, Edinburgh Napier University, Edinburgh EH105DT, UK; xiaodong.liu@napier.ac.uk (X.L.); g.vasantha@napier.ac.uk (V.G.)

³ School of Mechanical and Automotive Engineering, Xiamen University of Technology, Xiamen 361024, China; linwg@xmut.edu.cn

⁴ School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China

* Correspondence: rbxiao@hust.edu.cn

Abstract

Synonym similarity judgments based on semantic distance calculation play a vital role in supporting applications in the field of Natural Language Processing (NLP). However, existing semantic computing methods excessively rely on low-efficiency human supervision or high-quality datasets, which limits their further application. For these reasons, this paper proposes an automatic and intelligent method for calculating semantic similarity that integrates Subject–Action–Object (SAO) and WordNet to combine knowledge-based semantic similarity and corpus-based semantic similarity. First, the SAO structure is extracted from the Wikipedia dataset, and the statistics of SAO similarity are obtained by calculating co-occurrences of words in SAO. Second, the semantic similarity parameters of words are obtained based on WordNet, and the semantic similarity parameters are adjusted by Laplace Smoothing (LS). Finally, the semantic similarity can be obtained by the Bayesian Model (BM), which combines the semantic similarity parameter and the SAO similarity statistics. The experimental results from well-known word similarity datasets show that the proposed method outperforms traditional methods and even Large Language Models (LLM) in terms of accuracy. The Pearson, Spearman, and Kendall indices were introduced to prove the superiority of the proposed algorithm between model scores and human judgements.

Keywords: semantic similarity; subject–action–object; Bayesian Model; WordNet; Laplace Smoothing



Academic Editor: Vladimír Bureš

Received: 5 July 2025

Revised: 10 October 2025

Accepted: 10 October 2025

Published: 13 October 2025

Citation: Zeng, S.; Liu, X.; Lin, W.; Gokula, V.; Xiao, R. Integrated Subject–Action–Object and Bayesian Models of Intelligent Word Semantic Similarity Measures. *Systems* **2025**, *13*, 902. <https://doi.org/10.3390/systems13100902>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Word semantic similarity has always been the basis of text similarity analysis. The importance of word semantic similarity calculation is reflected in improving NLP application capabilities, such as statement similarity recognition, meaning disambiguation, semantic retrieval, human–machine interaction, and the quality of machine translation [1]. In the field of word semantic comparison, word similarity is mainly compared to word-formation characteristics and application context to understand the degree of similarity. In the field of word sense disambiguation (WSD), word similarity is required to identify the exact meaning of a word in context. In the field of machine translation, word similarity is mainly used to measure the interchangeability of the translated and original texts to accurately identify

the expression and obtain the translation. In the field of information retrieval, it enables users to accurately find the desired information, determine the scope of synonymous terms by identifying the words to be retrieved, and then expand the search results. In addition, semantic similarity has been widely used in structured resource clustering, geographic informatics, and biomedical science.

Semantic understanding has always been a challenge in NLP. The same concept may be expressed in different ways, and a word may contain multiple parts of speech, such as nouns and verbs. The semantic dilemma caused by multiple semantics has become a major obstacle in NLP. The key to settling the semantic dilemma is to solve the sensitive problem of synonymous concepts, that is, the similarity identification of the same semantics [2]. A set of high-quality semantic similarity recognition algorithms can effectively reduce the manual judgment workload and improve the efficiency of judgment. Among the current mainstream semantic similarity identification methods, the knowledge-based method relies on subjective experience, the corpus-based method has high requirements on the quality of the corpus, and the word vector-based method is difficult to reduce the complexity of the algorithm. To compensate for the shortcomings of a single method, semantic similarity calculation based on multi-method fusion has become mainstream in academia. However, existing research ignores the logic between the characteristics of words when measuring similarity, making it difficult to guarantee accuracy [3]. For example, the statements “motor drive belt” and “belt drive motor” logically belong to two concepts, but many research methods only consider co-occurrence as the judgment benchmark, resulting in the two sentences being judged to have the same concept.

SAO is triple-extracted from a text statement, in which the subject and object appear as the subjects and represent the executor and the executed in the event, and the predicate acts as an action between the two words as a structure based on part of speech and semantic acquisition. SAO can explain the logical relationship between words and avoid confusion. Using SAO as the information characteristic of words can avoid the interference of other irrelevant words in the context and accurately describe the associated information of words. Most importantly, it can reduce the workload of statistics and calculations and improve the accuracy of the semantic similarity algorithm.

In recent years, especially after Google launched its Transformer model, the efficiency of LLM has developed rapidly. LLM performs exceptionally well in various application scenarios, not only performing simple language tasks such as spell checking and grammar correction, but also handling complex tasks such as text summarization, machine translation, sentiment analysis, dialogue generation, and content recommendation. By pre-training on large-scale datasets, large language models acquire powerful general modeling capabilities and generalization capabilities. LLM also performs well in word’s semantic similarity tasks [4]. However, there are many limitations when targeting this fine-grained matching task [5]. First, although large language models are excellent at overall semantic understanding, in some cases, very subtle semantic differences are still difficult to accurately capture. Second, LLM typically requires a large amount of computational resources for inference and training, which can be a bottleneck when dealing with large-scale data or real-time applications. Third, the semantic understanding of LLM is highly dependent on contextual information, and the accuracy of semantic similarity judgment may decline for isolated words or scenes lacking context. Finally, if there is bias or imbalance in the training data, the model may perform poorly in judging the semantic similarity of words in specific domains or topics. Hence, LLM like a black box may encounter many problems while measuring word’s semantic similarity, even though these LLMs perform better in terms of computing results. By contrast, Explainable models such as BM are proposed to tackle the

inherent opacity in black-box systems, serving to help human users understand and trust machine learning outcomes.

It is conspicuous that word semantic similarity measurement is the basis of many prevalent high-level NLP tasks and LLM operations. In this context, to simultaneously address the limitations of existing research—namely, the oversight of logical relationships between lexical features and the lack of interpretability in the research process—this paper introduces an integrated word semantic similarity measurement model that fuses SAO and WordNet into the BM. First, SAO co-occurrence is utilized as the basis for determining word similarity to ensure the sequence of context features for guaranteeing logic order. Second, WordNet is applied to obtain semantic similarity parameters to avoid bias problems in corpus statistics. Third, LS is introduced to overcome the shortcoming of extreme circumstances within BM. Finally, a new similarity measurement based on BM is proposed: SAO co-occurrence statistics as SAO similarity, and the semantic similarity parameters of the knowledge base are incorporated to calculate the word semantic similarity. This method combines empirical and statistical semantic similarity based on experience. This research provides a novel idea of semantic similarity judgement and serves as a reference for improving judgmental accuracy and solving similarity calculation problems.

2. Related Works

The characteristics of words mainly include grammar, syntax, semantics, and pragmatics [6]. Semantic feature is the main component of semantic similarity calculation as the numerical expression of semantic similarity is one of the ways to measure the complex relationship between words [7]. Owing to cognitive differences, human-led semantic similarity results in the subjectivity of word similarity measurements. Therefore, based on objective practical experience, academia has summarized a variety of measurement methods, which are mainly divided into (i) knowledge-based methods, (ii) word vector-based methods, (iii) corpus-based methods, and (iv) multi-method fusion hybrid methods.

2.1. Knowledge-Based Semantic Similarity Calculation

To measure the semantic similarity between words, knowledge-based (or lexical and ontological knowledge bases) algorithms utilize semantic relationships (such as synonyms) and rules (such as hierarchies and adjacencies) within the knowledge base [8]. The lexical semantic knowledge base is a semantic dictionary, such as WordNet, MindNet, FrameNet, and Wiktionary.

Due to WordNet having strict structure and rules, many scholars choose it to calculate semantic similarity. For example, the WordNet synonym set was applied to WSD, extracting the relevant features by combining auxiliary languages and then disambiguating the meaning of words [9]. Three models were proposed to optimize the calculation based on the depth and shortest path between words in WordNet. Particle group optimization was also used to adjust the weights on the edge of the WordNet hierarchy to improve the correlation between words [10]. To improve the reliability of similarity estimation, it is necessary to accurately evaluate the information content in knowledge resources. Therefore, classification parameters were used to estimate the conceptual weight of information content [11]. To provide an innovative idea for the establishment and simulation of a combat system model, a WordNet-based hybrid semantic similarity calculation method was proposed, combined with semantic distance, ontology features, and feature parameter values [12]. Set algebra was applied to the WordNet synonym set by establishing general rules and using Boolean operators and lexical construction concept representation, and then the concept distribution was statistically analyzed to assign the most relevant meaning to the target words [13].

Additionally, Wikipedia-named entities were incorporated into the text as semantic representations, and the extended Naïve Bayes algorithm was introduced to eliminate noise, thereby reducing the calculation processing time [14]. Subsequent research has focused on category classification and conceptual information computing (IC) in Wikipedia, resulting in performance enhancements through improved IC measurement methods [15]. To address the data sparsity issue of Wikipedia, semantic information combination techniques were later employed, generating semantic description vectors for concept categories with notable success [16]. Similarly, the information neglect problem was resolved by integrating the IC and term frequency-inverse document frequency (TF-IDF) statistical models into a support vector machine (SVM) [17]. Furthermore, owing to the complexity of most methods and the challenges in effectively integrating WordNet and Wikipedia, an edge-weight model combining edge and density was proposed. This model converted Wikipedia links into semantic knowledge for integration with WordNet, thereby enhancing accuracy [18]. Based on FrameNet, a fuzzy SVM was applied to mine the semantic graph model, assigning weights to sentence nodes and edges, which eliminated redundant information and identified differences between documents [19].

HowNet is a concept represented by Chinese and English words as the description object to reveal the relationship between concepts and the relationship between the attributes of the concept. For instance, generalized suffix trees were used to extract common substrings of text, followed by a HowNet-based similarity calculation to form vocabulary links, ultimately enabling an accurate content-based search [20]. Subsequently, the importance of semantic similarity in geo-information science was emphasized, where geoscience terms were combined with HowNet to quantify the relationships between terms and construct a professional terminology database, addressing the challenges in similarity calculation for specialized terms [21]. In a similar approach, text was preprocessed using NLP, feature words were extracted using TF-IDF, and semantic similarity was calculated using HowNet [22].

Knowledge-based semantic similarity measurement requires a knowledge model (knowledge base), which, from the information content of the knowledge base, organizational structure, and characteristics of words, combines the algorithms of statistics, community algorithm, data structure, and network evaluation and fully excavates the internal structure of information in the knowledge base. Although this method enhances the accuracy of semantic similarity calculation, it lacks objectivity and has a higher requirement for the quality of the knowledge base model [23].

2.2. Semantic Similarity Calculation Method Based on the Corpus

Corpus in this case refers to a non-normative text corpus that differs from the lexical knowledge base, such as text downloaded or crawled from the Web, such as the public corpus on the Internet, or the contest corpus posted on NLP forums. These corpora are rich in information and diversified in semantic expression. The corpus-based semantic similarity calculation method uses context information to obtain the characteristics of target words and then calculates the semantic similarity of the words using statistical models [24].

To overcome the limitations of word-embedding-based methods, which overlook prior knowledge, and lexicon-based methods, which struggle with unknown words, a word-context network was extracted to construct the corpus, and a method was proposed to integrate prior knowledge obtained through vocabulary and statistics into word embeddings [25]. Additionally, word qualifiers were derived from Wikipedia to compute semantic similarity by comparing the qualifiers between words [26]. To enhance the WSD performance, an interactive range subgraph was constructed based on the context of ambiguous words from a public English WSD corpus. This was followed by similarity calculations

for ambiguous words, leading to significant improvements [27]. To calculate Chinese patent similarity, a link analysis algorithm was employed to assess the importance of each sequence structure, including functional and nonfunctional semantic relations [28]. Furthermore, a conceptual network was built using Wikipedia, and node distances were computed based on the access probabilities between nodes [29].

Since the corpus is usually automatically generated, it is easy to obtain and has good objectivity, which compensates for the error caused by subjective experience. However, the method based on a corpus is highly dependent on the distributional correlation of words, and the distribution correlation generated by different words is asymmetrical. In this case, a reasonable measurement method is needed, such as co-occurrence frequency, TF-IDF, set relevance, spatial distance, and retrieval feedback.

2.3. Semantic Similarity Calculation Method Based on Word Vectors

In recent years, with the rapid development of neural networks, an increasing number of studies have applied neural networks to measure semantic similarity. The semantic similarity calculation method based on word vectors is a distributed representation of words learned by a neural network, which represents the connection between words and context in vector form, and then the similarity calculation of words by the measurement between word vectors [30]. Typical word vector models include Word2Vec, GloVe, ELMo, and BERT. Corpus-based methods rely on statistical information (i.e., word frequency, co-occurrence matrix, mutual information, etc.) to obtain the semantic representation of words, and usually adopt sparse representation methods (such as word–word or word–context matrix). Correspondingly, the method based on word vectors uses neural network models to learn dense vector representations from large-scale texts, that is, mapping words to a low-dimensional real number vector, which can capture the potential semantic relationships between words.

A term–document matrix was constructed using a count-based method, and patent similarity was subsequently measured using Salton’s cosine metric [31]. A supervised classification approach was implemented, in which word pairs were initially represented by automatically extracted lexical pattern vectors, followed by SVM-based identification of given word pairs and subsequent algorithm optimization by testing different feature functions [32]. A lexicon–syntactic model was employed to define the vector space, utilizing cosine similarity to approximate the word semantic similarity. In this approach, each word is weighted according to its properties, with weights determined by their relationship to the central word. The improvement over traditional bag-of-words models was subsequently verified through clustering experiments [33]. To address the lack of research on semantic relevance within domain knowledge, singular value decomposition was applied to represent the word and topic vectors. This method, when combined with domain knowledge for topic similarity calculations, successfully identified prominent research target genes [34]. Finally, inspired by WordNet’s structural information, a novel word vector model was developed, incorporating three sub-models: a word definition model, word part-of-speech model, and word relation model [35].

Furthermore, recent studies have increasingly focused on the issue of semantic similarity in multilingual and multimedia environments. For example, Sviltana et al. [36] combined the hybrid model of n-gram statistics and semantic embedding to handle complex morphologies and was very effective in detecting various types of nearly repetitive texts. Zhang et al. [37] proposed an enhanced semantic similarity learning framework for image–text matching, which employs intra-modal multi-dimensional aggregators and inter-modal enhanced correspondence learning to dynamically capture richer semantic relations between visual and textual features, achieving state-of-the-art performance. Ismail et al. [38]

introduces an alignment word–space approach that integrates alignment-based and vector space-based methods for measuring semantic similarity between short texts, demonstrating improved performance on Arabic datasets with lemmatization and achieving one of the best reported accuracy levels. Muralikrishna et al. [39] proposed a method for measuring semantic similarity of Kannada–English sentence pairs by combining embedding space alignment, lexical decomposition, word order, and CNNs, achieving up to 83% correlation with human annotations and strong performance in semantic matching and retrieval tasks. Word vectors are more versatile; therefore, they have been favored by many NLP researchers. Meanwhile, most word vector models can be obtained through training, which reduces the manual work [40].

2.4. Hybrid Method Based on Multi-Method Fusion

Because of the advantages and disadvantages of each semantic similarity calculation, researchers have proposed a semantic similarity measurement method that integrates various methods. For example, the semantic similarity is calculated differently, and then the results are combined to obtain the average and maximum values. Another method takes different calculation models as part or parameters of the main model, which is called process fusion or embedded fusion.

To optimize semantic similarity calculation performance, the limitations of SVM-based approaches were first systematically analyzed, and an enhanced method was developed by combining the SVM's semantic similarity with word frequency functions derived from the nearest neighbor search, effectively reducing the dominance of high-frequency words and achieving results more consistent with human judgment [41]. Building on this foundation, lexical item spatial mapping was later extracted from WordNet and MeSH biomedical ontology, enabling the conversion of semantic similarity into part-of-speech spatial similarity, thereby improving the computational stability [42]. To enrich word vector semantics, a multi-semantic model was proposed that leverages WordNet's semantic attributes to refine corpus-based distributed representations, significantly enhancing the web service matching performance [43]. To address computational efficiency challenges, Word2Vec was employed to generate word vectors from corpus aggregates, with word sense expansion performed through WordNet, resulting in WSD accuracy surpassing baseline performance [44]. The EWS-CS model was subsequently introduced for semantic similarity evaluation, combining knowledge-based semantic features as knowledge encoders with word-embedding models as text encoders [45]. A comprehensive approach emerged through the integration of semantic dictionaries and corpora, featuring a holistic multi-relation framework that incorporates relationship-based information content and non-taxonomic weighted paths to measure both semantic similarity and relevance [46]. For patent analysis specifically, text similarity calculation was advanced through a novel BiGRU-HAN and GCN integration method that considers word co-occurrence probabilities [47]. Most recently, patent similarity assessment was further refined using Bi-LSTM+CRF models for function–object–property extraction combined with multiple matching methods including lexical-based, vector-based, and convolutional latent semantic approaches [48].

The semantic correlation calculation method based on multi-method fusion has become a research hotspot. The effectiveness of the semantic similarity measurement algorithm based on a single method has various shortcomings, whereas the multi-method fusion algorithm combines the advantages of different methods to further improve the effectiveness and make progress. However, these methods inevitably increase the computational complexity, and the compatibility between different models must be considered.

2.5. Drawbacks and Research Focus

As previously discussed, research on the calculation of semantic similarity of words generally takes the matching degree of word characteristics as the measurement standard, such as word semantic distance, contextual information, and text vector features. However, the existing research only considers the number of features when measuring features while ignoring properties. For example, in the calculation of context characteristics, only the words are counted, and the part of speech is rarely considered, resulting in poor precision. Although many studies annotated the part of speech in the semantic similarity calculation, only co-occurrence words of the same lexical context were counted, which separated the links between words in the text, resulting in inadequate information utilization.

The application of SAO in the judgment of keyword similarity is conducive to improving the accuracy of calculation because the greatest advantage of the SAO semantic network is that it labels words and extracts the structure of keywords in the text to ensure the logic between words. The method proposed in this paper refers to the idea of computing the semantic similarity BM, quantifying the properties (i.e., the depth of node and path distance) in WordNet, and then integrating it into the BM [49]. The method proposed in this study is based on a knowledge base. It obtains the semantic similarity probability from the keyword SAO in the corpus and combines it with the calculation method based on the information amount and distance of the path in the knowledge base. In addition, Laplacian smoothing was introduced to deal with the zero-probability problem.

3. Methodology

3.1. Overview

The semantic relations between words are mainly divided into three types of relations in Lexical Semantics: (i) hypernym relation, such as “fight” and “war,” which indicates that war is a form of fight; (ii) synonymous relation, for example, “car” and “automobile” are two different expressions of the same object; (iii) inclusion relation, for example, “handle” and “shower” represent the part and the whole item. Generally, two words in the above three relations have higher semantic similarity, whereas the antisense relationship is the opposite. By combining the BM based on the probability of SAO co-occurrence and the semantic relation of WordNet, the semantic similarity between words can be calculated effectively. This method can not only quantify the semantic proximity of words but also distinguish different types of semantic relations, thus improving the effectiveness and accuracy of NLP tasks.

3.2. Computing Model Design

Based on the shortcomings of the existing research and the main objectives of the semantic similarity algorithm, this study proposes a semantic similarity calculation model. The model fuses SAO and WordNet into the BM, which is divided into four parts. The specific implementation steps and BM building process are shown in Figure 1.

Step1: SAO extraction and data cleaning

First, the text in the corpus is segmented, the semantic and dependency of the words are labeled, and then the SAO structure is extracted according to the dependency parsing. Second, to compare the SAO structure of similar words, it is necessary to sum up the SAO obtained and then convert them into prototypes, since there are various lexical forms of words, such as the past, continuous, future, or plural forms. To facilitate statistics and make them easier to understand, the SAO structure of the words and their occurrence frequency in the text were counted.

Step2: SAO statistic similarity calculation

According to the statistical results obtained in Step1, when the two similar words are both Subjects, the SAO similarity is calculated according to the common actions and objects and their frequency, respectively. When both words are Actions, the SAO similarity is calculated according to the occurrence of Subjects and Objects and their frequency respectively. In the case where both the two comparison words are Objects, the SAO similarity is calculated according to the co-occurrence of the Subjects and Actions and their frequency, and if there is no co-existing SAO between word pairs, it results in 0 SAO similarity. LS was used to adjust the SAO similarity.

Step3: Experience similarity calculation

In this study, WordNet, which has more comprehensive and accurate information, was used as the knowledge base. By calculating the IC-based *Lin_similarity* and the concept-based *Res_similarity* of the two words, the highest score of both is obtained. The Brown corpus was chosen as the IC corpus. Finally, the mean of *Lin_similarity* and *Res_similarity* was calculated to obtain semantic similarity. Because semantic similarity is based on BM, when both the *Lin_similarity* and *Res_similarity* scores are 1, the semantic similarity is 0, which leads to the loss of the final BM similarity score. To solve this problem, the semantic similarity is adjusted by LS.

Step4: Combined semantic similarity calculation

According to the definition of the Bayesian theorem, if the SAO similarity score is taken as the probability of judging the similarity of the word pairs, the probability of not being similar is one minus the SAO similarity score. Considering the inevitable deviation, the SAO statistic similarity and WordNet semantic similarity was substituted into the Bayesian Formula to obtain the lexical similarity calculation results.

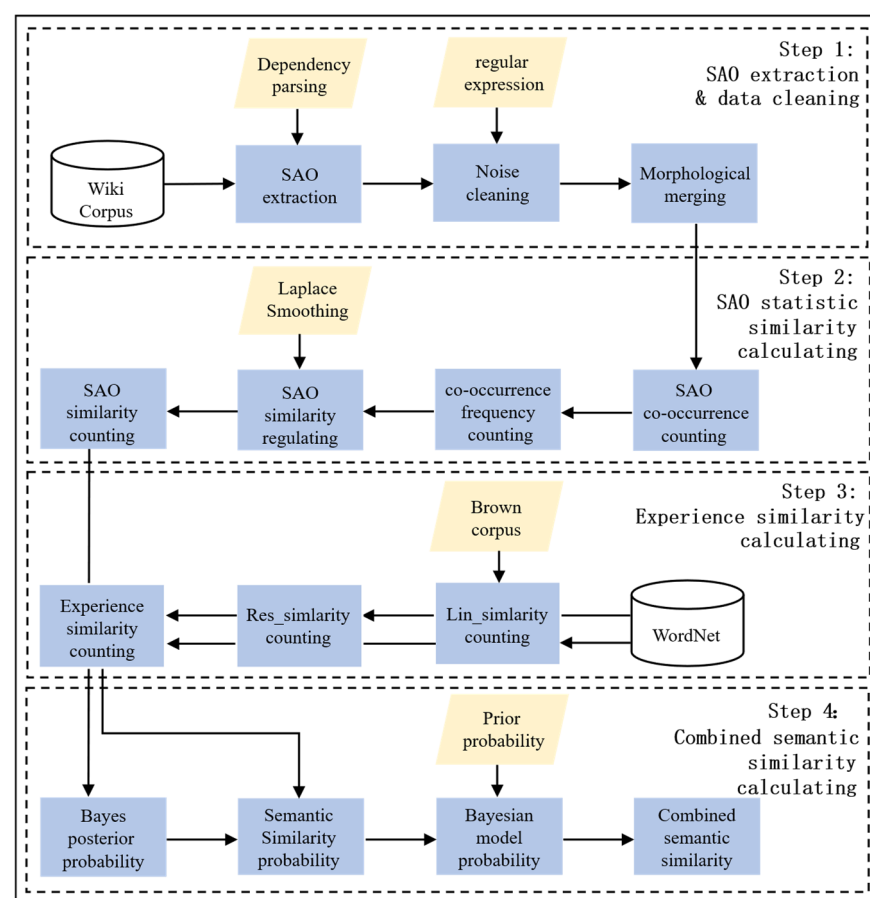


Figure 1. Flow chart of the model construction.

3.3. Similarity Calculation Based on SAO Statistics

Similarity calculation methods based on the corpus are mainly studied for the word context distribution, and if the context information of similar words is basically consistent, word similarity is calculated by the co-occurrence probability distribution of associative words in context. Commonly used methods for measuring the weight of context distribution information are term frequency (TF), inverse document frequency (IDF), TF-IDF, co-occurrence frequency, Boolean co-occurrence frequency, and mutual information [50]. Due to the different scope and habits of word usage, the number, frequency, and features of the words context in the corpus are different. The asymmetry of these characteristics limits the symmetry algorithm of cosine similarity and Jaccard similarity, and using word vector similarity computation can easily lead to dimensional disaster. In addition, the calculation method determines the quality of the semantic similarity calculation model. Thus, traditional semantic similarity calculation methods have many limitations and do not meet the demands of word semantic similarity calculation.

The SAO structure was selected in this study due to its balance between structural expressiveness and computational tractability. Unlike more complex representations such as Abstract Meaning Representation (AMR) or Semantic Role Labeling (SRL), which require large-scale annotated corpora and high-quality parsers, SAO extraction can be performed with relatively lightweight syntactic parsers and yields interpretable triplets that capture the fundamental propositional content of sentences. This makes SAO particularly advantageous for large-scale, unsupervised corpus analysis such as Wikipedia. Moreover, SAO emphasizes the logical relationality among lexical units by anchoring word semantics in functional roles (Subject, Action, Object), which mitigates ambiguity arising from shallow co-occurrence-based methods. Compared with dependency parse trees, SAO offers a normalized and flattened representation that abstracts away from syntactic variation (e.g., active vs. passive voice) and better supports alignment between conceptually equivalent expressions. In summary, the choice of SAO is motivated by its interpretability, scalability, and ability to bridge syntactic and semantic structures. While not as expressive as AMR or SRL, its relative simplicity and corpus adaptability make it suitable for the statistical semantic similarity task addressed in this work.

Based on the concept of information implication in information theory, this study proposes the concept of information implication similarity of the word SAO, which is defined as follows:

Definition 1. The SAO similarity of two words is determined by the quantity ratio of the SAO of the word with less information to the SAO of the word with more information.

Based on Definition 1, this study proposes the word SAO similarity calculation, as shown in Formula (1):

$$P_{(Ssim(w_k, w_l))} = \frac{P_{(w_{kS})} * P_{w_{lS}}}{P_{(w_{kS}, w_{lS})}} \text{ OR } P_{(Asim(w_k, w_l))} = \frac{P_{(w_{kA})} * P_{w_{lA}}}{P_{(w_{kA}, w_{lA})}} \text{ OR } P_{(Osim(w_k, w_l))} = \frac{P_{(w_{kO})} * P_{w_{lO}}}{P_{(w_{kO}, w_{lO})}} \quad (1)$$

where function $P_{(Ssim(w_k, w_l))}$ indicates the subject similarity of word k (w_k) and word l (w_l), while $P_{(Asim(w_k, w_l))}$ represents the predicate similarity and the $P_{(Osim(w_k, w_l))}$ represents the object similarity. $P_{(w_{kS})}$ is the probability of the co-occurrence of the subject of w_k and w_l ; similarly, $P_{w_{lS}}$ means the likelihood of the co-exist of the subject of w_l and w_k . $P_{(w_{kS}, w_{lS})}$ represents the sum of probability of the subject co-occurrence between w_k and w_l ; in a similar vein, $P_{(w_{kA}, w_{lA})}$ is the sum or probability of the action co-occurrence probability, and $P_{(w_{kO}, w_{lO})}$ is the sum or probability of the object co-occurrence probability. Given

w_k contains i groups of SAO in the corpus, while w_l has j groups of SAO in the corpus, Formulas (2)–(4) are shown below:

$$P_{(w_{ks})} = \frac{\sum w_{(k,l)S_i}}{\sum w_{kS_i}} \quad (2)$$

$$P_{(w_{ls})} = \frac{\sum w_{(l,k)S_i}}{\sum w_{lS_i}} \quad (3)$$

$$P_{(w_{ks},w_{ls})} = \frac{\sum w_{(k,l)S_i} + \sum w_{(l,k)S_i}}{\sum w_{kS_i} + \sum w_{lS_i}} \quad (4)$$

where $w_{(k,l)S_i}$ indicates the co-occurrence subject of w_k and w_l , $w_{(l,k)S_i}$ follows the same rule, w_{kS_i} is the total subjects of w_k , and w_{lS_i} is the total subjects of the w_l . The calculations of $P_{(w_{kA})}$, $P_{(w_{lA})}$, $P_{(w_{kO})}$, $P_{(w_{lO})}$, $P_{(w_{kA},w_{lA})}$, and $P_{(w_{kO},w_{lO})}$ follow the same analogy.

3.4. WordNet-Based Semantic Similarity Calculations

Wordnet is a widely used English semantic knowledge base in the NLP field, and it has a good conceptual hierarchy and tree-shaped structure. The semantic network is formed by the semantic relations between words, which is expressed through links. For example, the semantic relationship between the hyponym sets of the word “Channel” in WordNet is shown in Figure 2.

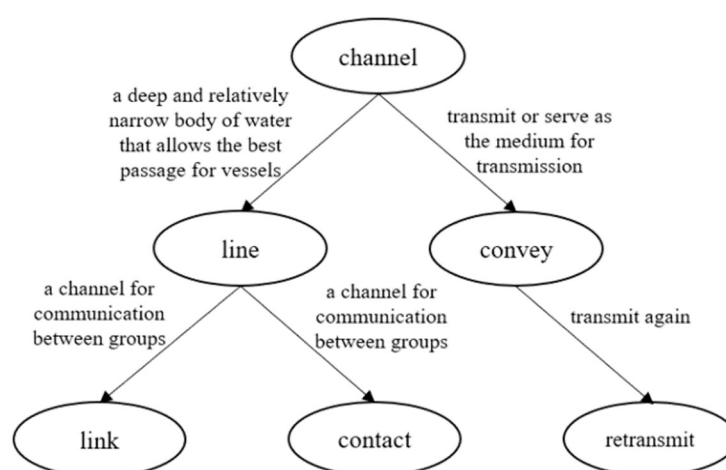


Figure 2. The structure of WordNet.

As can be seen from Figure 2, upper-level nodes are parent nodes of the lower-level one, which usually represents hyponym relation, and the nodes of the same parent node are sibling nodes, which are usually expressed as proximity or synonyms relation. For each definition, there is a definition of the word set that expresses the specific meaning.

WordNet was selected as the semantic resource for computing word similarity due to its well-established lexical ontology structure, which explicitly encodes multiple types of semantic relations, including synonymy, hyponymy, and meronymy. Compared to other semantic resources, WordNet offers several advantages:

(1). Hierarchical Organization: WordNet organizes nouns, verbs, adjectives, and adverbs into sets of cognitive synonyms, which are interlinked by conceptual–semantic and lexical relations. This allows for a systematic traversal of semantic hierarchies, which is essential for calculating semantic distances or similarities between concepts.

(2). Wide Adoption and Benchmarking: WordNet has become a de facto standard in lexical semantics and has been widely used as a benchmark resource for semantic

similarity research. Its structure supports various well-established similarity metrics, such as path-based, information content-based, and hybrid measures.

(3). Explicitly Defined Relations: Unlike corpus-derived embeddings that model semantics implicitly, WordNet provides explicitly defined and human-curated semantic relationships, which are particularly valuable for interpretability and explainability in similarity computation.

(4). Language and Domain Coverage: Although originally designed for general English, WordNet has been extended to other languages (e.g., EuroWordNet, Open Multilingual WordNet) and domains, enhancing its applicability in multilingual and domain-specific tasks.

(5). Compatibility with Hybrid Methods: WordNet can be integrated with statistical and neural methods to improve performance, enabling a hybrid approach that leverages both symbolic knowledge and distributional representations.

3.4.1. Semantic Similarity Calculation Based on Path Distance

The semantic similarity calculation model derived from WordNet is based on three aspects: (i) the path distance between words; (ii) the IC; and (iii) attribute features. Some studies consider using network model measurement algorithms to calculate word similarity, due to WordNet having a great network performance. The models are mainly determined by the number of connecting edges between two concepts. Since the number of joining edges is inversely proportional to semantic similarity and positively proportional to semantic distance, the more connected edges, the lower the semantic similarity and the farther the semantic distance. Path-based semantic similarity calculation methods have been enhanced through weighted edge assignments to improve computational accuracy. These methods primarily include four established approaches: (1) *Path_similarity* based on hierarchical relations; (2) *Wu_similarity* derived from an optimized shortest path algorithm that considers path length, conceptual taxonomy depth, and conceptual density [51]; (3) *LCS_similarity* developed from node depth measurement techniques [51]; and (4) *HSO_similarity* utilizing path-retrieval methodology. Among these, *Wu_similarity* demonstrates superior generality due to its comprehensive consideration of multiple semantic features.

3.4.2. Semantic Similarity Calculation Based on IC

WordNet has good network performance and also contains rich conceptual information. The IC-based semantic similarity calculation method mainly focuses on conceptual information. The foundational approach of this domain was first established through a vertex-based semantic similarity measurement method (*Res_similarity*), which computes concept similarity by determining the IC of the lowest common ancestor (LCA) between concepts [52]. The semantic similarity of concepts was evaluated according to the number of shared information concepts with common ancestors. This study selected *Res_similarity* as the IC-based semantic similarity method. The calculation of *Lin_similarity* is shown in Formula (5):

$$sim_{Res(w_k, w_l)} = \max_{c \in LCA(a, b)} IC(c), \quad IC(c) = -\log \frac{n(c)}{N} \quad (5)$$

where c is the LCA of a and b , and $n(c)$ is the number of c 's sub-concepts. N is the total number of concepts in the ontology. To normalize the *Res_similarity*, Formula (5) is transformed into Formula (6):

$$sim_{Res(w_k, w_l)} = \frac{\max_{c \in LCA(a, b)} IC(c)}{\max_{c \in LCA(a, a)} IC(c)} \quad (6)$$

Moreover, based on *Res_similarity*, Jiang & Conrath proposed *Jcn_similarity* of the nearest ancestor node of the combined word. Lin et al. (1998) proposed the *Lin_similarity*

semantic similarity calculation model by taking the ratio of information required to express commonality between words and the amount of information required to fully describe words as a measure of the similarity between concepts [53]. Since of Lin's method quantifies the results within the range of [0, 1] and fully considers the commonality and IC of each concept, this study selected *Lin_similarity* method as another IC-based semantic similarity experience score. The calculation of *Lin_similarity* is shown in Formula (7):

$$sim_{Lin(w_k, w_l)} = \frac{2 * \lg P(dcp(w_k, w_l))}{\lg P(w_k) + \lg P(w_l)} = \frac{2 * IC(dcp(w_k, w_l))}{IC(w_k) + IC(w_l)} \quad (7)$$

where $P(dcp(w_k, w_l))$ represents the probability of the proportion of information of the nearest common parent node of w_k and w_l to the total description information, and $P(w_k)$ represents the probability of the proportion of information of w_k to the total information. $P(w_l)$ follows the same rule as $P(w_k)$. As shown in Figure 2, the nearest common parent node of "link" and "contact" is "line", and then the *Lin_similarity* of both is the ratio of twice the information amount of "line" to the sum of both information amount.

3.4.3. Total Experience Similarity Calculation

There are always problems with a single WordNet similarity algorithm, thus, the idea of computing semantic similarity by fusing two algorithms is presented in this paper.

Based on the results of the above empirical scoring algorithms, the empirical similarity calculation of words based on knowledge based is shown in Formula (8), which takes w_k and w_l as examples.

$$sim_{exp} = \alpha sim_{Lin} + \beta sim_{Res} \quad (8)$$

where α and β are the adjustment factors, and $\alpha + \beta = 1$.

3.5. Semantic Similarity Calculation Based on BM

The BM algorithm is a simple classification algorithm based on reversion thinking, Bayesian theorem, and the assumption of independent characteristic conditions. The classic applications of BM include sentiment classification model, spam filtering model, and spell checker.

3.5.1. LS Adjustment

Since the computational semantic similarity in this paper was based on the BM, zero probability events will lead to extreme results (0 or 1). LS was used to adjust the zero probability events to avoid extreme result. Formulas (9) and (10) are the calculations for LS:

$$P_{\lambda}(X^{(j)} = a_{jl} | Y = c_k) = \frac{\sum_{i=1}^N I(x_i^{(j)} = a_{jl}, y_i = c_k) + \lambda}{\sum_{i=1}^N I(y_i = c_k) + s_j \lambda} \quad (9)$$

$$P_{\lambda}(Y = c_k) = \frac{\sum_{i=1}^N I(y_i = c_k) + \lambda}{N + K \lambda} \quad (10)$$

where λ represents the smoothing parameter, which is generally set to 1 to avoid 0 in molecule. a_{jl} represents the observation of the value 1 under the j -th feature result of random variable a . k is the number of feature species, s_j represents the number of feature classification, and N is the number of statistical samples.

There are two cases in which LS needs to be introduced: (i) when the empirical similarity score is 1; (ii) SAO similarity does not exist in the same situation.

When the words are under the same parent node, the similarity empirical score is 1 and the dissimilarity empirical score is 0. However, on the basis of BM, the introduction of empirical score resulted in the semantic calculation is completely dependent on SAO

similarity, which is obviously against the original intention: the experience similarity of knowledge base and SAO work together. Therefore, Laplacian smoothing is introduced in the case where the semantic similarity empirical score is 1. The score can be divided into two categories, namely $s_j = 2$, set the number of statistical samples $N = 10$, $\lambda = 1$, the number of statistically similar samples is adjusted for similarity = $10 + 1$, dissimilarity = 1, $P((w_k, w_l)|similarity) = \frac{10+1}{10+2}$, and $P((w_k, w_l)|dissimilarity) = \frac{0+1}{10+2}$. LS adjustment ensures the accuracy of the algorithm, which avoids the situation of losing function when the score is 0.

In the statistics of co-occurrence SAO, especially when the words belong to the anti-sense relationship or ontology relationship of different domains, no co-occurrence SAO situations frequently happen. In this situation, once $P_{(w_{kS}, w_{lS})}$, $P_{(w_{kA}, w_{lA})}$, and $P_{(w_{kO}, w_{lO})}$ are 0, the result of the whole BM is also 0, which results in failure. To avoid that, we introduced LS to adjust the SAO co-occurrence statistics. We divided the original results into two categories, co-occurrence and non-co-occurrence, that is, $s_j = 2$, and set $\lambda = 1$:

$$P_{(w_{kS}, w_{lS})|similarity} = \frac{\sum w_{(k,l)A_i} + \sum w_{(l,k)A_j} + 1}{\sum w_{kA_i} + \sum w_{lA_i} + 2} = \frac{0 + 1}{T} \quad (11)$$

$$P_{(w_{kS}, w_{lS})|dissim} = \frac{\sum w_{(k,l)A_i} + \sum w_{(l,k)A_j} + 1}{\sum w_{kA_i} + \sum w_{lA_i} + 2} = \frac{B + 1}{T} \quad (12)$$

where T represents the sum of statistical samples and B indicates the total of statistical samples not contributed. After the adjustment, the statistical co-occurrence probability of SAO being 0 is avoided, which removes the obstacles for the similarity calculation of BM and improves the accuracy of the algorithm.

3.5.2. BM for Word Similarity Calculation

The higher the semantic similarity between words, the greater the probability of substitution. The fundamental cause of the substitution is that the SAO semantic generality between words without large bias or ambiguity. For example, in the sentence “I have a car” and “I have an automobile”, “car” and “automobile” mean the same thing. In addition, the subject “I” and the predicate “have” remain the same, thus, the substitution of car and automobile will not change the original meaning. Through analysis, the greater the frequency of SAO co-occurrence between words, the more opportunities for replacement, and the higher the semantic similarity. Therefore, this paper proposed to take the SAO statistical model as the SAO similarity probability of the BM, as shown in Figure 3, and take the similarity experience score based on the knowledge base as the semantic consistency parameter, and finally obtain the results of the similarity calculation of words and words.

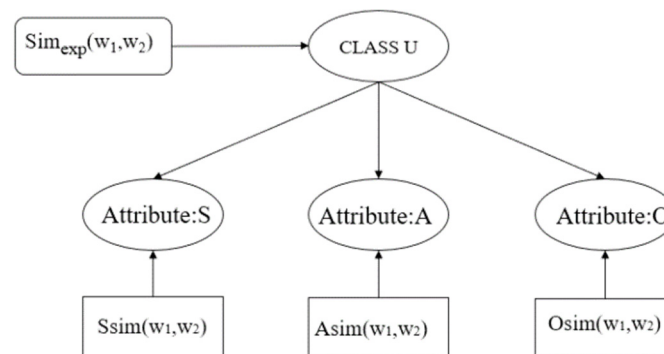


Figure 3. The semantic similarity of words to BM.

where $Ssim(w_1, w_2)$ is the similarity score of attribute Subject (the calculated words as subject) of words, $Asim(w_1, w_2)$ is the similarity score of attribute Action (the calculated words as action) of words, and $Osim(w_1, w_2)$ is the similarity score of attribute Object (the calculated words as object) of words. Class U represents the judgment result, range within $\{T, F\}$, T means “similar”, and F means “dissimilar”. Assuming that marginal distribution is $P(Y)$, then $P(Ssim | Y)$, $P(Asim | Y)$, and $P(Osim | Y)$ represent the probability distribution of semantic similarity between word pairs. In the study of [43], the prior probability $P(Y = T)$ and $P(Y = F)$ were set to 0.5, and then the posterior probability of judging similar words based on SAO similarity was $P(Y = T | Ssim)$, $P(Y = T | Ssim)$, and $P(Y = T | Ssim)$. The posterior probability of judging that the words are not similar was $P(Y = T | Ssim)$, $P(Y = T | Ssim)$, and $P(Y = T | Ssim)$. Assuming that w_k and w_l are subjects, the semantic similarity under the joint constraints of attribute $Asim(w_k, w_l)$ and attribute $Osim(w_k, w_l)$ is $sim(w_k, w_l)$. The calculation is shown in Formula (13):

$$sim(w_k, w_l) = \frac{sim_{exp}(w_k, w_l) * P(Y = T | Asim(w_k, w_l), Osim(w_k, w_l))}{sim_{exp}(w_k, w_l) * P(Y = T | Asim(w_k, w_l), Osim(w_k, w_l)) + Z} \quad (13)$$

where Z represents the joint probability that w_k and w_l are not similar, and the calculation is shown in Formula (14):

$$Z = (1 - sim_{exp}(w_k, w_l)) * P(Y = F | Asim(w_k, w_l), Osim(w_k, w_l)) \quad (14)$$

Obtained from the conditional probability formula, the calculations of $P(Y = T | Asim(w_k, w_l), Osim(w_k, w_l))$ and $P(Y = F | Asim(w_k, w_l), Osim(w_k, w_l))$ are shown in Formulas (15) and (16):

$$P(Y = T | Asim(w_k, w_l), Osim(w_k, w_l)) = \frac{P(Y = T, Asim(w_k, w_l), Osim(w_k, w_l))}{P(Asim(w_k, w_l), Osim(w_k, w_l))} \quad (15)$$

$$P(Y = F | Asim(w_k, w_l), Osim(w_k, w_l)) = \frac{P(Y = F, Asim(w_k, w_l), Osim(w_k, w_l))}{P(Asim(w_k, w_l), Osim(w_k, w_l))} \quad (16)$$

Obtained from the joint probability formula, the calculation of $P(Y = T, Asim(w_k, w_l), Osim(w_k, w_l))$ is shown in Formula (17):

$$P(Y = T, Asim(w_k, w_l), Osim(w_k, w_l)) = P(Y = T) * P(Asim(w_k, w_l) | Y = T) * P(Osim(w_k, w_l) | Y = T, Asim(w_k, w_l)) \quad (17)$$

For $P(Osim(w_k, w_l) | Y = T, Asim(w_k, w_l))$, since $Osim(w_k, w_l)$ and $Asim(w_k, w_l)$ are mutually exclusive, $P(Osim(w_k, w_l) | Y = T, Asim(w_k, w_l))$ is simplified to Formula (18):

$$P(Osim(w_k, w_l) | Y = T, Asim(w_k, w_l)) = P(Osim(w_k, w_l) | Y = T) \quad (18)$$

Incorporate Formula (18) into Formula (17), from which Formula (15) is simplified to Formula (19):

$$\frac{P(Y = T, Asim(w_k, w_l), Osim(w_k, w_l))}{P(Y = T) * P(Asim(w_k, w_l) | Y = T) * P(Osim(w_k, w_l) | Y = T)} = \frac{P(Y = T) * P(Asim(w_k, w_l) | Y = T) * P(Osim(w_k, w_l) | Y = T)}{P(Asim(w_k, w_l), Osim(w_k, w_l))} \quad (19)$$

Similarly, $P(Y = F, Asim(w_k, w_l), Osim(w_k, w_l))$ and $P(Osim(w_k, w_l) | Y = F, Asim(w_k, w_l))$ are simplified to Formulas (20) and (21):

$$P(Y = F, Asim(w_k, w_l), Osim(w_k, w_l)) = P(Y = F) * P(Asim(w_k, w_l) | Y = F) * P(Osim(w_k, w_l) | Y = F, Asim(w_k, w_l)) \quad (20)$$

$$P(Osim(w_k, w_l) | Y = F, Asim(w_k, w_l)) = P(Osim(w_k, w_l) | Y = F) \quad (21)$$

Combine $P(Y = T) = P(Y = F) = 0.5$ with Formulas (19)–(21) to simplify Formula (13) to Formula (22):

$$\text{sim}(w_k, w_l) = \frac{\text{sim}_{\text{exp}}(w_k, w_l) * P(\text{Asim}(w_k, w_l)) * P(\text{Osim}(w_k, w_l))}{\text{sim}_{\text{exp}}(w_k, w_l) * P(\text{Asim}(w_k, w_l)) * P(\text{Osim}(w_k, w_l)) + X} \quad (22)$$

where X represents the joint probability that w_k and w_l are not similar, and the calculation is shown in Formula (23):

$$X = (1 - \text{sim}_{\text{exp}}(w_k, w_l)) * (1 - P(\text{Asim}(w_k, w_l))) * (1 - P(\text{Osim}(w_k, w_l))) \quad (23)$$

In addition, if w_k and w_l are Actions, semantic similarity is calculated from the joint constraint of attributes $\text{Ssim}(w_k, w_l)$ and $\text{Osim}(w_k, w_l)$, respectively. When w_k and w_l are Objects, then they are calculated from the joint constraint of attributes $\text{Ssim}(w_k, w_l)$ and $\text{Asim}(w_k, w_l)$.

4. Experimental Results and Analysis

The text used in this study was a public corpus of Wikipedia. As an encyclopedia of all areas of knowledge, Wikipedia has thousands of entries and provides the most detailed answers possible. Wikipedia data was collected in the Wikimedia website¹. A total of more than 1.8 million articles were obtained from the Wikipedia corpus. After analyzing semantically similar word pairs in the database, it was observed that, when two subjects exhibit high semantic similarity, their corresponding actions and objects also tend to exhibit high similarity. Conversely, when two subjects are semantically dissimilar, the associated actions and objects likewise demonstrate low semantic similarity. Thus, the SAO-based word meaning similarity measurement was developed.

The experimental environment was configured as follows: (Intel(R) Xeon(R) Gold 5218R CPU @ 2.10GHz) *2, 64.00 GB RAM, Windows10 64-bit operating system, Python 3 version programming language.

4.1. Extraction and Statistics of SAO Structure

Based on the existing Wikipedia corpus, the pure text extracted in the corpus was more than 25 GB. Due to the data being large, which enlarges the burden of the machine, only one-tenth of text (by date from front to back) was used to be the dataset. The size of the dataset reduces the probability of accidents and improves the reliability of the database. Firstly, the sentence including keywords was extracted from the dataset, and then the SAO structure was extracted in these sentences via Dependency Parsing; the Spacy module was introduced as the parser. The pseudo code for SAO extraction process is described in Table 1, as well as the time complexity of the algorithm $T(n) = O(n^2)$. From a complexity perspective, the proposed method maintains a manageable computational overhead. SAO extraction involves syntactic parsing per sentence, with worst-case complexity $O(n^2)$, where n is the number of tokens. WordNet lookup operations are constant time under index-based access. In contrast, transformer-based baselines (e.g., BERT) require $O(n^3)$ self-attention operations and heavy GPU resources. Hence, even without parallelism, our method is more suitable for lightweight or batch processing contexts. Finally, the SAO structure of Wikipedia text is shown in Table 2.

Table 1. The pseudo code for SAO extraction process.

```

01. // Variable: PT— —provided text
02. // Variable: ST— —sentence
03. // Variable: WD— —word
04. // Variable: WDPOS— —word with POS tagging
05. // Function: SAO— —subject-action-object
06. // Function: POST— —POS tagging
07. // Function: DEP_GRAM— —dependency grammar
08. // Variable: DataFrame— —distributed data set
09. Begin:
10. import nlp package
11. import CSV to DataFrame
12. Subject_POS_list = ["Subj", "nsubj", "nsubjpass"]
13. Action_POS_list = ["aux", "auxpass", "complem", "prt", "cmod", "mmod"]
14. Object_POS_list = ["obj", "dobj", "pobj"]
15. def getSubject (ST):
16.     for word in ST:
17.         WD = word
18.         WDPOS = POST(word)
19.         If WDPOS in Subject_POS_list:
20.             Subject = WD
21.     Return Action
22. def getAction (ST):
23.     for word in ST:
24.         WD = word
25.         WDPOS = POST(word)
26.         If WDPOS in Action_POS_list:
27.             Action = WD
28.     Return Action
29. def getObject (ST):
30.     for word in ST:
31.         WD = word
32.         WDPOS = POST(word)
33.         If WDPOS in Object_POS_list:
34.             Object = WD
35.     Return Object
36. data = pd. DataFrame()
37. for i in range(len(csv)):
38.     ST = split (PT, Regular expression clause)
39.     for j in range (len (ST)):
40.         Subject = getSubject (WD)
41.         Action = getAction (WD)
42.         Object = getObject (WD)
43.         SAO = (Subject,Action,Object)
44.     end
45.     data = data.append(SAO)
46. end for.
47. output(data)

```

The proposed method relies on dependency parsing for SAO extraction, which may introduce errors for complex syntactic constructions such as nested clauses, long-distance dependencies, and passive voice. To estimate the potential impact, a sample of 200 sentences containing relative clauses and passive constructions from the Wikipedia subset was manually annotated by two experts; one is a doctor of English linguistics, and the other is a professor in the field of NLP. The SpaCy version 2 parser achieved approximately 91.2% precision and 88.7% recall for subject, verb, and object relations in these sentences. It is nearly equal to the benchmarks of accuracy that are mentioned on the website of

SpaCy (<https://spacy.io/> accessed on 1 October 2025). Most errors were due to attachment ambiguities (e.g., prepositional phrase attachment) and auxiliary verb misclassification in passive voice. While these errors can slightly affect SAO co-occurrence counts, the BM partially mitigates their effect by combining statistical evidence with WordNet-based similarity. Future work will explore using Transformer-based dependency parsers (e.g., Stanza, SpaCy-trf) or hybrid syntactic–semantic parsing approaches, as well as parser-confidence weighting, to further improve SAO extraction robustness.

Table 2. SAO structure of the Wikipedia text.

Subject	Action	Object
technique	be	diffusion
technique	be	society
technique	be	automobile
firm	call	name
firm	call	dispute
.....		
state	emerge	company
state	emerge	Leyland
state	emerge	district
he	announce	average
he	announce	automobile

This study selected word pairs to compare and extract the SAO structure of the words according to their grammatical forms (i.e., Subject, Action, or Object). Some words have multiple forms, such as prototype, plural, and so on, which require conversion to the prototype. Firstly, we conducted word-form conversion of WordNet Lemmatizer in the Python NLTK database. Then, the SAO of similar words was extracted, the total number of SAO was counted by Formula (4), and the co-occurrence SAO was marked. Finally, the SAO co-occurrence was counted by Formula (11), and the SAO similarity was obtained by Formula (22). At the same time, to find its knowledge base-based semantic similarity score, the Python's WordNet module was applied to calculate the Lin and Res between the word pairs and calculate the empirical similarity of the two algorithms by Formula (8). From Formula (7), the value of α and β was undefined. To prove the best combination of α and β , the different configuration of α and β was defined to calculate the semantic similarity. The correlation indicators under different covariates are shown in Figure 4.

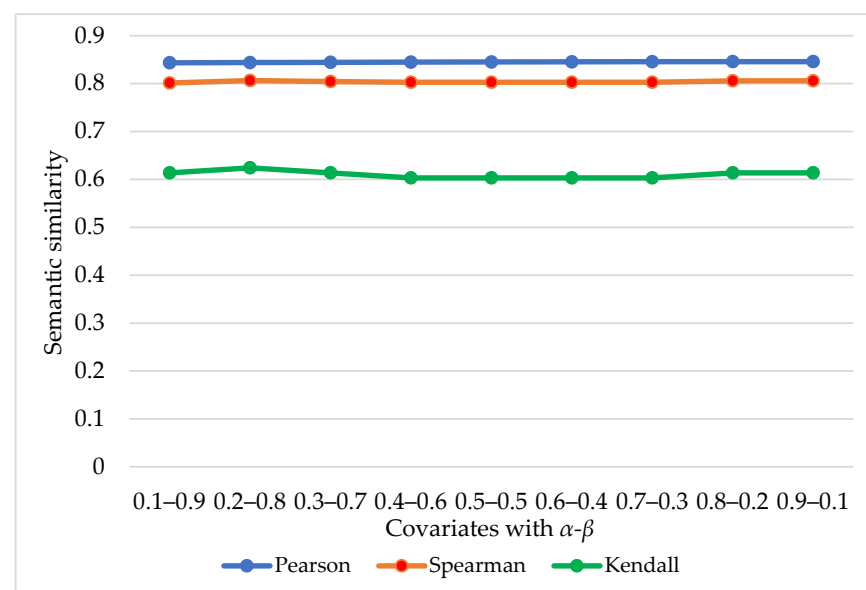


Figure 4. Correlation indicators under different covariates.

As reflected in Figure 4, it is obvious that the performance of the model is better when the adjustment factor set α as 0.2 and β as 0.8 compared to other sets; the final configuration of α and β is specified. The calculation results of SAO similarity between word pairs and empirical similarity based on knowledge base are shown in Table 3.

Table 3. SAO statistic similarity calculation results and semantic similarity calculation results.

w_k	w_l	$P_{(Asim(w_k, w_l))}$	$P_{(Osim(w_k, w_l))}$	sim_{exp}	$sim(w_k, w_l)$
cord	smile	0.452857	0.13441	0.051275	0.006898
rooster	voyage	0.628738	0.190753	0.083333 *	0.035019
noon	string	0.889726	0.487714	0.060029	0.329107
fruit	furnace	0.802228	0.275732	0.217836	0.300741
autograph	shore	0.108461	0.721767	0.083333	0.02789
forest	graveyard	0.877965	0.299686	0.128814	0.312819
food	rooster	0.678134	0.33227	0.089671	0.093606
cemetery	woodland	0.836161	0.388748	0.118864	0.304518
shore	voyage	0.701818	0.322306	0.083333 *	0.092363
bird	woodland	0.844491	0.551971	0.139818	0.520956
crane	implement	0.683133	0.260719	0.306226	0.25127
brother	lad	0.896143	0.458434	0.250188	0.709062
sage	wizard	0.669027	0.273837	0.190021	0.1517
oracle	sage	0.67659	0.272945	0.5117	0.451457
bird	crane	0.882723	0.453146	0.762029	0.952318
cushion	pillow	0.601014	0.089431	0.988341	0.926149
cemetery	graveyard	0.896422	0.539231	0.916667 *	0.991104
automobile	car	0.872531	0.558889	0.916667 *	0.989627
midday	noon	0.846892	0.475569	0.916667 *	0.982199
gem	jewel	0.663112	0.184116	0.916667 *	0.830107

Note: * presents the results are adjusted by LS.

Finally, the SAO statistic similarity and semantic empirical score were fused into the BM to obtain the word pair similarity value. Forty pairs of words were selected for verification, and the final empirical score and SAO similarity calculation are shown in the “ $sim(w_k, w_l)$ ” column of Table 3.

To enhance the evaluation of the algorithm’s effectiveness, the same sampling method was used to verify the WordSim-353 dataset, and the verification results are shown in Table 4. By comparing with the manual scoring in the WordSim-353 dataset, we verified the calculation results of the model proposed in this paper using three metrics: Pearson, Spearman, and Kendall. The verification results are as follows: Pearson coefficient is 0.874207, Spearman coefficient is 0.761394, and Kendall coefficient is 0.674169, which thereby demonstrates the reliability of the model proposed in this paper across different datasets.

Table 4. Semantic similarity calculation results of the WordSim-353 database.

w_k	w_l	$P_{(Asim(w_k, w_l))}$	$P_{(Osim(w_k, w_l))}$	sim_{exp}	$sim(w_k, w_l)$	WordSim-353
cord	smile	0.452857	0.13441	0.051275	0.006898	0.08125
rooster	voyage	0.628738	0.190753	0.083333 *	0.035019	0.09375
noon	string	0.889726	0.487714	0.060029	0.329107	0.06875
tiger	zoo	0.325918	0.561597	0.125942	0.300741	0.4.9375
tiger	carnivore	0.413865	0.786143	0.788223	0.02789	0.6
forest	graveyard	0.877965	0.299686	0.128814	0.312819	0.1394
food	rooster	0.678134	0.33227	0.089671	0.093606	0.25625
cemetery	woodland	0.836161	0.388748	0.118864	0.304518	0.23125
crane	implement	0.683133	0.260719	0.306226	0.25127	0.125
brother	lad	0.896143	0.458434	0.250188	0.709062	0.49375
seafood	lobster	0.471582	0.728424	0.809234	0.451457	0.69375
bird	crane	0.882723	0.453146	0.762029	0.952318	0.48125
money	currency	0.68243	0.812571	0.863502	0.991104	0.79375
automobile	car	0.872531	0.558889	0.916667 *	0.989627	0.9369
midday	noon	0.846892	0.475569	0.916667 *	0.982199	0.89375
gem	jewel	0.663112	0.184116	0.916667 *	0.830107	0.8375

Note: * presents the results are adjusted by LS.

4.2. Algorithm Comparison and Evaluation

Correlation coefficient is used to measure of the strength of the relationship between two variables. Specifically, the correlations are computed between the model-generated similarity values and the reference similarity scores assigned by human subjects. This allows us to quantitatively evaluate how well the proposed algorithm aligns with human semantic intuition. In statistics, the main correlation coefficients used are Pearson and Spearman. The Pearson correlation coefficient mainly measures the degree of linear correlation between the two variables, the value of which is in the range of $[-1, 1]$. The closer the correlation coefficient is to 1, the more linear correlation between the variables, the closer the correlation coefficient is to -1 , the more linear negative correlation, and 0 indicates no correlation. Spearman correlation coefficient mainly uses monotony function to describe the degree of relationship between two variables, and its calculation method is mainly based on the ranking of variables rather than the original data. The calculations of Pearson and Spearman correlation coefficients are shown in Formulas (24) and (25):

$$\gamma_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2} \sqrt{n\sum y_i^2 - (\sum y_i)^2}} \quad (24)$$

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2 - 1)} \quad (25)$$

where γ_{xy} represents the Pearson correlation coefficient between variable x and y , and n represents the observation times; ρ represents the Spearman correlation coefficient, and d_i^2 represents the level difference of the corresponding variable.

In this research, R&G 65 was selected to be the dataset and the value was normalized [54]. Considering the vast workload, we used stratified sampling to select 40 pairs of words for semantic similarity calculation and comparison. It is acknowledged that the current evaluation was based on a subset of 40-word pairs, which may appear limited in scope. However, this subset was not randomly or arbitrarily selected; instead, a systematic sampling strategy was employed from the R&G 65 dataset to ensure representative distribution across similarity levels. Specifically, the dataset was evenly divided into five parts within the range from 0 to 1, that is, each 0–0.2 similarity was regarded as a subset, and 8-word pairs were sampled at intervals for semantic similarity comparison. This sampling approach mitigates selection bias by preserving structural diversity in semantic distance. Moreover, the R&G 65 dataset has been widely adopted as a standard benchmark in the evaluation of word-level semantic similarity, where studies often use partial sampling due to the high cost of human-annotated references. The partial results of this method are shown in Table 5 below.

Table 5. Semantic similarity calculation results based on different models.

w_k	w_l	sim_{exp}	$sim(w_k, w_l)$	Word2Vec	BERT-Base	Sentence BERT	GPT-4o	Sentence Transformer	ESimCSE	R&G 65
cord	smile	0.051275	0.006898	0.2094	0.0134	0.2812	0.3903	0.18	0.7688	0.005
rooster	voyage	0.083333 *	0.035019	0.0324	0.1711	0.6917	0.3604	0.161	0.5536	0.01
noon	string	0.060029	0.329107	0.0485	0.1656	0.6283	0.0448	0.24	0.6767	0.01
fruit	furnace	0.217836	0.300741	0.2184	0.1794	0.7350	0.1771	0.261	0.5415	0.0125
autograph	shore	0.083333	0.02789	0.1452	0.0784	0.4136	0.1567	0.183	0.5580	0.015
forest	graveyard	0.128814	0.312819	0.1316	0.2057	0.4616	0.1202	0.32	0.5585	0.25
food	rooster	0.089671	0.093606	0.0770	0.2225	0.6904	0.9702	0.324	0.7085	0.2725
cemetery	woodland	0.118864	0.304518	0.3355	0.2971	0.3921	0.1103	0.761	0.6295	0.295
shore	voyage	0.083333 *	0.092363	0.3525	0.3431	0.6920	0.9978	0.393	0.6346	0.305
bird	woodland	0.139818	0.520956	0.3726	0.6901	0.6558	0.7305	0.289	0.6277	0.31
crane	implement	0.306226	0.25127	0.1239	0.5324	0.2793	0.9046	0.208	0.6460	0.5925
brother	lad	0.250188	0.709062	0.2395	0.6634	0.5579	0.9996	0.524	0.8659	0.6025
sage	wizard	0.190021	0.1517	0.3087	0.9087	0.7114	0.9984	0.317	0.6071	0.615
oracle	sage	0.5117	0.451457	0.4334	0.8811	0.6731	0.9996	0.271	0.6373	0.6525
bird	crane	0.762029	0.952318	0.4770	0.8912	0.4617	0.9846	0.436	0.6850	0.6575
cushion	pillow	0.988341	0.926149	0.4355	0.7802	0.6318	0.9993	0.608	0.7876	0.96
cemetery	graveyard	0.916667 *	0.991104	0.5928	0.7704	0.4617	0.9993	0.848	0.9383	0.97
automobile	car	0.916667 *	0.989627	0.6300	0.8441	0.6860	0.9838	0.865	0.9492	0.98
midday	noon	0.916667 *	0.982199	0.7986	0.9143	0.7825	0.9997	0.809	0.9065	0.985
gem	jewel	0.916667 *	0.830107	0.7570	0.9049	0.4584	0.9998	0.66	0.8160	0.985

Note: * presents the results are adjusted by LS.

While modern LLMs (e.g., GPT-4, DeepSeek-R1) exhibit remarkable performance in sentence-level semantic tasks, their applicability to fine-grained word-level similarity assessment remains fundamentally limited. State-of-the-art LLMs are generally fine-tuned to generate high-quality representations at the sentence or document level, and their embeddings often capture contextual semantics holistically, making them less suitable for isolated or pairwise word-level similarity tasks. Moreover, obtaining word-level embeddings from these models typically involves additional complexity—such as attention-weighted token aggregation or probing intermediate layers—none of which are standardized or directly comparable across models. In order to analyze the performance of each module in the proposed model, an ablation study was conducted in this experiment, and the results are shown in Table 6. To ensure a fair comparison with the proposed method, a word-level similarity calculation baseline based on BERT was designed. Specifically, the BERT-base-uncased model from HuggingFace’s Transformer library was adopted. For each target word, the input was marked using WordPiece tokenization, and its contextualization representation was obtained from the final hidden layer. When a word is divided into multiple sub-word labels, the average of all sub-word vectors is calculated to represent the word. Then, the similarity between the two words is measured by calculating the cosine similarity between their final average embeddings.

Table 6. Results of the ablation study and model comparison.

Index \ Module	P_Asim	P_Osim	WordNet	BERT-Base	SentenceBERT	ESimCSE	GPT-4o	BM	R&G 65
Spearman	0.327028	0.00366	0.72753	0.90254	−0.02958	0.81400	0.78390	0.845241	1
Pearson	0.234011	0.10083	0.79676	0.69473	0.00264	0.76119	0.83216	0.80286	1
Kendall	0.137568	0.09524	0.63618	0.80424	−0.00180	0.60589	0.62130	0.603183	1

Where “P_Asim” represents action similarity of word pairs, “P_Osim” represents objective similarity, “WordNet” denotes semantic similarity derived from WordNet, and “BM” refers to the semantic similarity method proposed in this paper. Table 6 compares the proposed method with traditional approaches, earlier pre-trained models (BERT-base, Sentence-BERT), contrastive models (ESimCSE), and the latest large language model GPT-4o. The results show that the proposed BM method consistently achieves high correlation with the R&G 65 benchmark. For Spearman correlation, BM reaches 0.845241, outper-

forming WordNet (0.72753), Sentence-BERT (0.81400), ESIMCSE (0.78390), and GPT-4o (0.78390), while being slightly lower than BERT-base (0.90254). For Pearson correlation, BM obtains 0.802860, surpassing BERT-base (0.69473), Sentence-BERT (0.76119), and ESIMCSE (0.76119), and closely approaching GPT-4o (0.83216). For Kendall correlation, BM achieves 0.603183, comparable to other strong baselines and showing robustness across different correlation measures. These three correlation coefficients verified that proposed method has better performance.

To explicitly quantify the contribution of the SAO component, we compared the performance of using WordNet-based similarity alone against the proposed SAO+WordNet BM. As shown in Table 6, the Spearman correlation improves from 0.72753 (WordNet-only) to 0.84524 with the inclusion of SAO co-occurrence statistics, representing an absolute gain of 11.8%. Similarly, the Pearson correlation increases from 0.79676 to 0.80286. These results clearly demonstrate that the SAO structure significantly enhances semantic similarity estimation by capturing contextual and relational information beyond what is encoded in WordNet alone. These results highlight the key advantages of the proposed BM method: it combines high accuracy, strong stability, and computational efficiency while avoiding the complexity and large-scale training requirements of recent pre-trained models. BM thus provides a concise, interpretable, and practically deployable solution for semantic similarity measurement, demonstrating competitive performance even against state-of-the-art language models like GPT-4o. In practice, the use of LLM to calculate word semantic similarity is not effective; maybe, it is mainly because they are designed to generate natural language and understand context, rather than focusing on generating high-quality word vectors. This results in poor semantic consistency of the embedding vectors, while the lack of context may obscure the meaning of polysemous words. In addition, the vector space may be distorted, the model's understanding of words in a particular domain is limited, and the way semantic similarity is evaluated may not fully reflect human semantic perception.

Cosine similarity on distributional embeddings (e.g., Word2Vec, BERT) captures contextual usage patterns of words, which can be advantageous for polysemy resolution in rich textual environments. However, such embeddings are inherently black-box representations whose semantic axes are latent and not directly interpretable. Moreover, embeddings trained on large corpora often smooth out fine-grained lexical distinctions: near-synonyms and antonyms appearing in similar contexts may be assigned similar vectors, leading to inflated similarity scores. In contrast, the proposed SAO-WordNet-BM approach explicitly encodes structural and ontological relations between words. WordNet provides curated synonymy, hypernymy, and meronymy links, which capture semantic categories rather than distributional co-occurrence patterns. The SAO component further anchors words in their grammatical and functional roles, enabling context reconstruction without requiring full sentence embedding.

The result of Table 3 is based on Lin of Formula (6) and *Res_similarity* of Formula (7), for verify the effect of combining the two algorithms to calculate the semantic similarity of SAO, the correlation indicators under different WordNet semantic similarity algorithms is shown as Table 7.

Table 7. Correlation indicators under different the WordNet algorithm.

Algorithm	Pearson	Spearman	Kendall
path-lin	0.752761	0.738902	0.560854
path-wup	0.646398	0.646351	0.455033
path-res	0.752834	0.735892	0.550272
path-lch	0.625761	0.671182	0.507944
lin-wup	0.756309	0.741911	0.560854
lin-res	0.845241	0.80286	0.603183
lin-lch	0.721924	0.727615	0.550272
wup-res	0.758635	0.741911	0.560854
wup-lch	0.641655	0.659895	0.476197
res-lch	0.722661	0.727615	0.550272

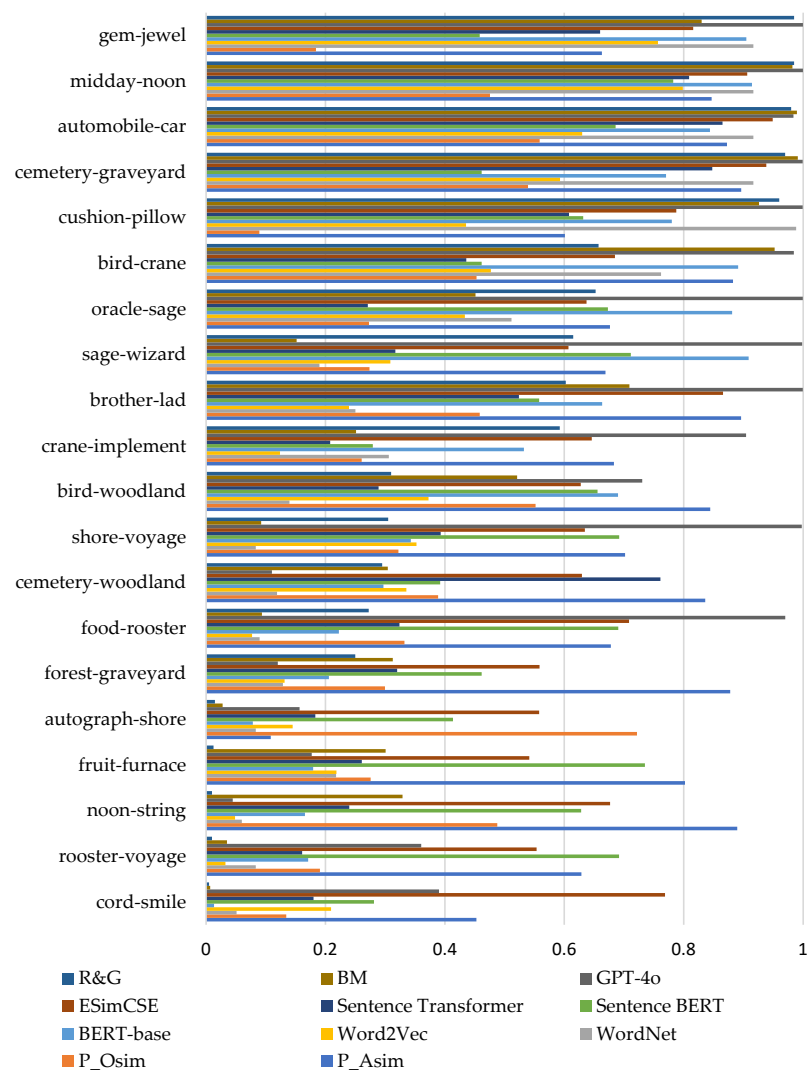
The Ordinary Least Squares (OLS) summary report is a detailed output that provides various metrics and statistics to help evaluate the model's performance and interpret its results [55]. It is important to note that the purpose of this analysis is not to establish causal relationships, but rather to assess the degree of linear agreement between the predicted and reference similarity values. Specifically, this study treated model-predicted similarity as the independent variable and human-annotated similarity as the dependent variable, and fit a univariate linear regression model. This approach enables to interpret both the regression slope and the coefficient of determination (R^2) as indicators of alignment fidelity: as the slope (RG parameter) closes to 1, a high F-statistic value and R-squared value suggest strong linear consistency between our method and the human benchmark. While OLS is often used for causal inference in econometric studies, it is also widely adopted in computational linguistics and evaluation studies for curve fitting and model calibration purposes, especially when the focus is on comparative consistency rather than explanatory causality. Understanding each one can reveal valuable insights into the model's performance and accuracy. Hence, OLS was introduced in this research, and the report is shown in Table 8.

According to the OLS report, the linear correlation coefficient (shown as RG index) reaches 0.835149 and the F-statistic reaches 2.72×10^{-6} , and R-squared is 0.699; these parameters show that the method applied in this paper has a strong configuration and significance effect with the manual scoring results. Based on the data shown in Table 5 and the scoring results of manual annotation, the comparison results of various algorithms are shown in Figure 5, where 'WordNet' is the similarity score based on WordNet, 'BM' is the proposed method, 'Word2Vec' is the similarity score based on Word2Vec by training the Wikipedia dataset, and 'RG' is the word pairs of R&G 65. As shown in Figure 5, the method proposed in this article is closer to the results of R&G 65 than others. Furthermore, Mean Squared Error (MSE) is introduced to measure the difference between the predicted values of the model and the true values. It evaluates the prediction performance of the model by calculating the square meaning of the prediction error. The smaller the MSE, the more accurate the model's prediction. Conversely, the larger the MSE, the greater the error of the model. After calculating, the MSE value between proposed model and R&G 65 dataset is 0.040346, which indicates the effectiveness of the proposed model.

Table 8. OLS report.

Intercept	0.066876					
RG	0.835149					
dtype: float64						
OLS Regression Results						
=====						
Dep. Variable:	BM	R-squared:	0.714			
Model:		OLS	Adj. R-squared:	0.699		
Method:		Least Squares	F-statistic:	45.03		
Date:	Sun, 4 September 2022	Prob (F-statistic):	2.72×10^{-6}			
Time:	17:13:34	Log-Likelihood:	4.6857			
No. Observations:	20	AIC:	−5.371			
Df Residuals:	18	BIC:	−3.380			
Df Model:	1					
Covariance Type:	nonrobust					
=====						
coef	std err	t	P > t	[0.025	0.975]	

Intercept	0.0669	0.074	0.900	0.380	−0.089	0.223
RG	0.8351	0.124	6.711	0.000	0.574	1.097
=====						
Omnibus:		0.740	Durbin-Watson:			2.315
Prob(Omnibus):		0.691	Jarque-Bera (JB):			0.634
Skew:		−0.387	Prob(JB):			0.728
Kurtosis:		2.600	Cond. No.			3.45
=====						

**Figure 5.** Comparison results of different calculation methods.

4.3. Discussion

The proposed model introduces a novel integration of syntactic structure and probabilistic reasoning for measuring semantic similarity. By incorporating SAO triples into a Bayesian framework, this study extends beyond traditional co-occurrence and path-based similarity methods by embedding logical and contextual structures into semantic evaluation. The definition of information implication similarity provides a new metric grounded in the asymmetry and informativeness of SAO co-occurrence patterns, reflecting a cognitively inspired approach to lexical comparison. Furthermore, by introducing LS to mitigate the zero-probability problem inherent in sparse data or extreme empirical scores, the model ensures robustness and reliability in similarity estimation. This multi-source fusion framework contributes to the theoretical development of interpretable, structured, and statistically sound similarity modeling.

In practical terms, the proposed method offers a lightweight and explainable alternative to computationally intensive models such as large-scale transformers. It is particularly suitable for fine-grained semantic similarity tasks where explainability and transparency are required, such as synonym detection, entity alignment, domain-specific knowledge extraction, and patent document analysis. Its compatibility with open resources like Wikipedia and WordNet enables its deployment in low-resource or multilingual environments. Unlike black-box models that struggle with isolated word pairs or ambiguous terms lacking context, the proposed method directly encodes grammatical roles and semantic relations, which enhances its applicability in industrial NLP pipelines, including semantic search, question answering systems, and legal or biomedical text analytics.

Cosine similarity on distributional embeddings (e.g., Word2Vec, BERT) captures contextual usage patterns of words, which can be advantageous for polysemy resolution in rich textual environments. However, such embeddings are inherently black-box representations whose semantic axes are latent and not directly interpretable. Moreover, embeddings trained on large corpora often smooth out fine-grained lexical distinctions: near-synonyms and antonyms appearing in similar contexts may be assigned similar vectors, leading to inflated similarity scores.

In contrast, the proposed SAO–WordNet–BM approach explicitly encodes structural and ontological relations between words. WordNet provides curated synonymy, hypernymy, and meronymy links, which capture semantic categories rather than distributional co-occurrence patterns. The SAO component further anchors words in their grammatical and functional roles, enabling context reconstruction without requiring full sentence embeddings. The experimental results (shown in Table 4) confirm that our method produces higher correlation with human judgments on synonym pairs (e.g., “automobile–car,” “midday–noon”) compared with Word2Vec cosine similarity, suggesting that it better preserves lexical semantic equivalence. Furthermore, the Bayesian combination mitigates the “flattening” effect by weighting SAO statistics and WordNet knowledge, thus avoiding excessive smoothing and improving sensitivity to subtle distinctions.

One potential concern is the reliance of the proposed method on WordNet, whose development has slowed down in recent years. While WordNet’s core lexical ontology is considered mature and stable, its lack of frequent updates may limit coverage for emerging terms and domain-specific neologisms. To address this, it is possible that the WordNet component can be replaced by or complemented with alternative lexical resources such as BabelNet, ConceptNet, or domain-specific ontologies (e.g., UMLS for biomedical terms) without altering the Bayesian integration mechanism. Furthermore, the SAO co-occurrence statistics are corpus-driven and can be continuously updated as new text data becomes available, ensuring that the model remains adaptive to linguistic evolution. Thus, the

proposed method is expected to remain applicable in the long term and can be extended to leverage multilingual or dynamically updated lexical databases.

As the Wikipedia corpus is crowd-sourced, it inevitably contains grammatical noise, non-standard sentence constructions, and occasional spelling errors, which may lead to inaccurate SAO extraction. For example, run-on sentences and ungrammatical phrasing can mislead dependency parsing, while spelling variants can fragment word statistics. To mitigate these issues, we applied basic preprocessing steps including sentence segmentation, lowercasing, and token normalization. Additionally, the large scale of the Wikipedia corpus statistically dilutes the effect of isolated noise, and the Bayesian integration with WordNet similarity helps stabilize the final similarity scores. Nevertheless, we acknowledge that corpus noise may still contribute to residual errors. Future work will incorporate grammar-checking tools, spelling correction, and outlier filtering to further improve the quality of SAO co-occurrence statistics and reduce the impact of noisy data on similarity estimation.

5. Conclusions

In this study, the shortcomings of existing semantic similarity calculation methods are summarized, and a multi-method fusion semantic similarity calculation method is proposed to reduce the complexity of time and to improve the accuracy of the calculation results. This paper introduced the model of SAO information based on the common statistics of the SAO words in the corpus, which can better solve the mismatch between the number and dimension of conceptual feature sets, and further improve the information utilization of the corpora-based semantic similarity computing method. Then, WordNet was used as the knowledge base and the relevant semantic similarity calculation method to calculate the empirical similarity score. Moreover, the results of the SAO similarity model were adjusted and integrated into BM. The Bayesian posterior probability between concepts was taken as the final calculation result of the semantic similarity. The feasibility of this method was verified by comparing, discussing, and analyzing the method in this paper with other algorithms and manual judgment. Overall, this work provides a scalable and interpretable alternative for semantic similarity computation, with both academic and applied implications.

The main contributions of this study can be summarized as follows: Novel Hybrid Framework—An integrated semantic similarity model that fuses SAO co-occurrence statistics with WordNet-based semantic similarity is proposed, forming a BM that jointly exploits syntactic structure and lexical knowledge. Information Implication Similarity—The concept of information implication similarity is introduced, which quantifies asymmetry between SAO co-occurrence patterns and improves fine-grained similarity measurement. Robustness via LS-LS is applied to address zero-probability problems in sparse SAO data, enhancing model stability and reliability. Efficiency and Interpretability—The proposed method achieves competitive accuracy with lower computational complexity than transformer-based methods, making it suitable for large-scale and low-resource scenarios, while remaining explainable and interpretable. Empirical Validation—Experimental results on the R&G 65 benchmark demonstrate that the proposed approach outperforms traditional methods and even surpasses large language models in word-level similarity tasks.

It is important to clarify that this study does not aim to introduce a fundamentally new semantic theory of meaning representation. Rather, our focus is on methodological innovation—designing a practical, interpretable, and statistically principled framework for word-level semantic similarity. By systematically integrating SAO co-occurrence patterns with WordNet-based semantic relations within a Bayesian probabilistic model, the proposed approach bridges structural knowledge and statistical inference. This combination not only improves prediction accuracy but also enhances explainability, a key requirement in

XAI applications. Although we do not claim to redefine the essence of semantic similarity, our work provides an operationally effective and theoretically consistent tool that can be directly applied to synonym detection, entity alignment, and semantic search in real-world NLP systems. This contribution complements existing theoretical research by enabling scalable, interpretable deployment of semantic similarity computation.

Author Contributions: Conceptualization, S.Z., X.L., and R.X.; methodology, W.L.; software, W.L. and X.L.; validation, W.L.; formal analysis, S.Z. and W.L.; investigation, S.Z. and W.L.; resources, W.L.; data curation, X.L.; writing—original draft preparation, S.Z., W.L., and X.L.; writing—review and editing, W.L., V.G., and S.Z.; visualization, W.L. and X.L.; supervision, S.Z., V.G., and W.L.; project administration, S.Z., W.L., and R.X.; funding acquisition, S.Z., W.L., and R.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under grant No. 52275249, and the Social Science Foundation of Fujian Province under grant No. FJ2023B077.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Note

- ¹ <https://dumps.wikimedia.org/enwiki/latest/> (accessed on 22 March 2025)

References

- Chandrasekaran, D.; Mago, V. Evolution of Semantic Similarity—A Survey. *ACM Comput. Surv.* **2021**, *54*, 1–37. [\[CrossRef\]](#)
- Yan, F.; Fan, Q.; Lu, M. Improving Semantic Similarity Retrieval with Word Embeddings. *Concurr. Comput. Pract. Exp.* **2018**, *30*, e4489. [\[CrossRef\]](#)
- Wang, X.; Zhu, D.; Yang, C. SAO Semantic Information Identification for Text Mining. *Int. J. Comput. Intell. Syst.* **2017**, *10*, 593–604. [\[CrossRef\]](#)
- Liu, A.; Pan, L.; Lu, Y.; Li, J.; Hu, X.; Zhang, X.; Wen, L.; King, I.; Xiong, H.; Yu, P. A Survey of Text Watermarking in the Era of Large Language Models. *ACM Comput. Surv.* **2024**, *57*, 1–36. [\[CrossRef\]](#)
- Manakul, P.; Liusie, A.; Gales, M.J.F. Selfcheckgpt: Zero-resource Black-box Hallucination Detection for Generative Large Language Models. *arXiv* **2023**, arXiv:2303.08896.
- Özçelik, Ö. Interface Hypothesis and the L2 Acquisition of Quantificational Scope at the Syntax-semantics-pragmatics Interface. *Lang. Acquis.* **2017**, *25*, 213–223. [\[CrossRef\]](#)
- Gao, J.-B.; Zhang, B.-W.; Chen, X.-H. A WordNet-based Semantic Similarity Measurement Combining Edge-counting and Information Content Theory. *Eng. Appl. Artif. Intell.* **2015**, *39*, 80–88. [\[CrossRef\]](#)
- Chergui, O.; Ahlame, B.; Dominique, G.-L. Integrating a Bayesian Semantic Similarity Approach into CBR for Knowledge Reuse in Community Question Answering. *Knowl.-Based Syst.* **2019**, *185*, 104919. [\[CrossRef\]](#)
- Lopez-Arevalo, I.; Sosa-Sosa, V.J.; Rojas-Lopez, F.; Tello-Leal, E. Improving Selection of Synsets from WordNet for Domain-Specific Word Sense Disambiguation. *Comput. Speech Lang.* **2017**, *41*, 128–145. [\[CrossRef\]](#)
- Cai, Y.; Zhang, Q.; Lu, W.; Che, X. A Hybrid Approach for Measuring Semantic Similarity Based on IC-weighted Path Distance in WordNet. *J. Intell. Inf. Syst.* **2017**, *51*, 23–47. [\[CrossRef\]](#)
- Zhang, X.; Sun, S.; Zhang, K. An Information Content-Based Approach for Measuring Concept Semantic Similarity in WordNet. *Wirel. Pers. Commun.* **2018**, *103*, 117–132. [\[CrossRef\]](#)
- Ensor, T.M.; MacMillan, M.B.; Neath, I.; Surprenant, A.M. Calculating Semantic Relatedness of Lists of Nouns Using WordNet Path Length. *Behav. Res. Methods* **2021**, *53*, 2430–2438. [\[CrossRef\]](#)
- Azad, H.-K.; Akshay, D. A New Approach for Query Expansion Using Wikipedia and WordNet. *Inf. Sci.* **2019**, *492*, 147–163. [\[CrossRef\]](#)
- Nakamura, T.; Shirakawa, M.; Hara, T.; Nishio, S. Wikipedia-Based Relatedness Measurements for Multilingual Short Text Clustering. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2018**, *18*, 1–25. [\[CrossRef\]](#)
- Jiang, Y.; Bai, W.; Zhang, X.; Hu, J. Wikipedia-based Information Content and Semantic Similarity Computation. *Inf. Process. Manag.* **2017**, *53*, 248–265. [\[CrossRef\]](#)
- Qu, R.; Fang, Y.; Bai, W.; Jiang, Y. Computing Semantic Similarity Based on Novel Models of Semantic Representation using Wikipedia. *Inf. Process. Manag.* **2018**, *54*, 1002–1021. [\[CrossRef\]](#)

17. Wasti, S.H.; Hussain, M.J.; Huang, G.; Akram, A.; Jiang, Y.; Tang, Y. Assessing Semantic Similarity between Concepts: A Weighted-feature-based Approach. *Concurr. Comput. Pract. Exp.* **2020**, *32*, e5594. [\[CrossRef\]](#)
18. Li, D.; Summers-Stay, D. Dual Embeddings and Metrics for Relational Similarity. *Ann. Math. Artif. Intell.* **2019**, *88*, 533–547. [\[CrossRef\]](#)
19. Wang, T.; Shi, H.; Liu, W.; Yan, X. A joint FrameNet and element focusing Sentence-BERT method of sentence similarity computation. *Expert Syst. Appl.* **2022**, *200*, 117084. [\[CrossRef\]](#)
20. Fu, X.; Liu, W.; Xu, Y.; Cui, L. Combine HowNet Lexicon to Train Phrase Recursive Autoencoder for Sentence-Level Sentiment Analysis. *Neurocomputing* **2017**, *241*, 18–27. [\[CrossRef\]](#)
21. Chen, Z.; Song, J.; Yang, Y. An Approach to Measuring Semantic Relatedness of Geographic Terminologies Using a Thesaurus and Lexical Database Sources. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 98. [\[CrossRef\]](#)
22. Pang, S.; Yao, J.; Liu, T.; Zhao, H.; Chen, H. A Text Similarity Measurement Based on Semantic Fingerprint of Characteristic Phrases. *Chin. J. Electron.* **2020**, *29*, 233–241. [\[CrossRef\]](#)
23. Araque, O.; Zhu, G.; Iglesias, C.A. A Semantic Similarity-Based Perspective of Affect Lexicons for Sentiment Analysis. *Knowl.-Based Syst.* **2019**, *165*, 346–359. [\[CrossRef\]](#)
24. Coelho, J.; Mano, D.; Paula, B.; Coutinho, C.; Oliveira, J.; Ribeiro, R.; Batista, F. Semantic Similarity for Mobile Application Recommendation under Scarce User Data. *Eng. Appl. Artif. Intell.* **2023**, *121*, 105974. [\[CrossRef\]](#)
25. Huang, D.; Pei, J.; Zhang, C.; Huang, K.; Ma, J. Incorporating Prior Knowledge into Word Embedding for Chinese Word Similarity Measurement. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2018**, *17*, 1–21. [\[CrossRef\]](#)
26. Zhu, X.; Yang, X.; Huang, Y.; Guo, Q.; Zhang, B. Measuring Similarity and Relatedness Using Multiple Semantic Relations in WordNet. *Knowl. Inf. Syst.* **2019**, *62*, 1539–1569. [\[CrossRef\]](#)
27. Kwon, S.; Oh, D.; Ko, Y. Word Sense Disambiguation based on Context Selection using Knowledge-based Word Similarity. *Inf. Process. Manag.* **2021**, *58*, 102551. [\[CrossRef\]](#)
28. An, X.; Li, J.; Xu, S.; Chen, L.; Sun, W. An Improved Patent Similarity Measurement based on Entities and Semantic Relations. *J. Informetr.* **2021**, *15*, 101135. [\[CrossRef\]](#)
29. Hussain, M.J.; Bai, H.; Jiang, Y. Wikipedia bi-linear link (WBLM) model: A New Approach for Measuring Semantic Similarity and Relatedness between Linguistic Concepts using Wikipedia Link Structure. *Inf. Process. Manag.* **2023**, *60*, 103202. [\[CrossRef\]](#)
30. Lastra-Díaz, J.J.; Goikoetxea, J.; Hadj Taieb, M.A.; García-Serrano, A.; Ben Aouicha, M.; Agirre, E. A Reproducible Survey on Word Embeddings and Ontology-based Methods for WordSimilarity: Linear Combinations Outperform the State of the Art. *Eng. Appl. Artif. Intell.* **2019**, *85*, 645–665. [\[CrossRef\]](#)
31. Chen, D.-Z.; Chou, C.-H. Exploring Technology Evolution in the Solar Cell Field: An Aspect from Patent Analysis. *Data Inf. Manag.* **2017**, *1*, 124–134. [\[CrossRef\]](#)
32. Li, H.; Xu, Z.; Li, T.; Sun, G.; Raymond Choo, K.-K. An Optimized Approach for Massive Web Page Classification using Entity Similarity based on Semantic Network. *Future Gener. Comput. Syst.* **2017**, *76*, 510–518. [\[CrossRef\]](#)
33. Spasic, I.; Corcoran, P.; Gagarin, A.; Buerki, A. Head to Head: Semantic Similarity of Multi-word Terms. *IEEE Access* **2018**, *6*, 20545–20557. [\[CrossRef\]](#)
34. Henry, S.; Cuffy, C.; McInnes, B.T. Vector Representations of Multi-word Terms for Semantic Relatedness. *J. Biomed. Inform.* **2018**, *77*, 111–119. [\[CrossRef\]](#) [\[PubMed\]](#)
35. Zhao, F.; Zhu, Z.; Han, P. A Novel Model for Semantic Similarity Measurement based on Wordnet and Word Embedding. *J. Intell. Fuzzy Syst.* **2021**, *40*, 9831–9842. [\[CrossRef\]](#)
36. Biloshchytska, S.; Arailym, T.; Oleksandr, K.; Andrii, B.; Yurii, A.; Sapar, T.; Aidos, M.; Saltanat, S. Text Similarity Detection in Agglutinative Languages: A Case Study of Kazakh Using Hybrid N-Gram and Semantic Models. *Appl. Sci.* **2025**, *15*, 6707. [\[CrossRef\]](#)
37. Kun, Z.; Hu, B.; Zhang, H.-T.; Li, Z.; Mao, Z.-D. Enhanced Semantic Similarity Learning Framework for Image-Text Matching. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 2973–2988.
38. Muralikrishna, S.N.; Raghurama, H.; Harivinod, N.; Raghavendra, G. Cross-Lingual Short-Text Semantic Similarity for Kannada-English Language Pair. *Computers* **2024**, *13*, 9.
39. Ismail, S.; Tarek, E.L.-S.; Abdelwahab, K.-A. A New Alignment Word-Space Approach for Measuring Semantic Similarity for Arabic Text. *Int. J. Semant. Web Inf. Syst.* **2022**, *18*, 1–18. [\[CrossRef\]](#)
40. Biggers, F.B.; Mohanty, S.D.; Manda, P. A Deep Semantic Matching Approach for Identifying Relevant Messages for Social Media Analysis. *Sci. Rep.* **2023**, *13*, 12005. [\[CrossRef\]](#)
41. Hussain, S.F.; Suryani, A. On Retrieving Intelligently Plagiarized Documents using Semantic Similarity. *Eng. Appl. Artif. Intell.* **2015**, *45*, 246–258. [\[CrossRef\]](#)
42. Lastra-Díaz, J.J.; Lara-Clares, A.; Garcia-Serrano, A. HESML: A real-time Semantic Measures Library for the Biomedical Domain with A Reproducible Survey. *BMC Bioinform.* **2022**, *23*, 23. [\[CrossRef\]](#)

43. Batet, M.; Sánchez, D. Leveraging Synonymy and Polysemy to Improve semantic Similarity Assessments based on Intrinsic Information content. *Artif. Intell. Rev.* **2019**, *53*, 2023–2041. [[CrossRef](#)]
44. Orkphol, K.; Yang, W. Word Sense Disambiguation Using Cosine Similarity Collaborates with Word2vec and WordNet. *Future Internet* **2019**, *11*, 114. [[CrossRef](#)]
45. Yin, F.; Wang, Y.; Liu, J.; Ji, M. Enhancing Embedding-Based Chinese Word Similarity Evaluation with Concepts and Synonyms Knowledge. *Comput. Model. Eng. Sci.* **2020**, *124*, 747–764. [[CrossRef](#)]
46. AlMousa, M.; Benlamri, R.; Khoury, R. Exploiting Non-taxonomic Relations for Measuring Semantic Similarity and Relatedness in WordNet. *Knowl.-Based Syst.* **2021**, *212*, 106565. [[CrossRef](#)]
47. Chen, L.; Xu, S.; Zhu, L.; Zhang, J.; Yang, G.; Xu, H. A Deep Learning based Method Benefiting from Characteristics of Patents for Semantic Relation Classification. *J. Informetr.* **2022**, *16*, 101312. [[CrossRef](#)]
48. Teng, H.; Wang, N.; Zhao, H.; Hu, Y.; Jin, H. Enhancing Semantic Text Similarity with Functional Semantic Knowledge (FOP) in Patents. *J. Informetr.* **2024**, *18*, 101467. [[CrossRef](#)]
49. Shirakawa, M.; Kotaro, N.; Takahiro, H.; Shojiro, N. Wikipedia-Based Semantic Similarity Measurements for Noisy Short Texts Using Extended Naive Bayes. *IEEE Trans. Emerg. Top. Comput.* **2015**, *3*, 205–219. [[CrossRef](#)]
50. Rahutomo, F.; Aritsugi, M. Econo-ESA in Semantic Text Similarity. *Springerplus* **2014**, *3*, 149. [[CrossRef](#)] [[PubMed](#)]
51. Wang, J.H.; Zuo, W.L.; Yan, Z. Word Semantic Similarity Measurement based on Naive Bayes Model. *J. Comput. Res.* **2015**, *52*, 1499–1509.
52. Resnik, P. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In Proceedings of the IJCAI'95: Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, QC, Canada, 20–25 August 1995; pp. 448–453.
53. Lin, D. An Information-theoretic Definition of Similarity. In Proceedings of the ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning, Madison, WI, USA, 24–27 July 1998; Volume 98, pp. 296–304.
54. Rubenstein, H.; Goodenough, J.B. Contextual correlates of synonymy. *Commun. ACM* **1965**, *8*, 627–633. [[CrossRef](#)]
55. Du, J.; Park, Y.T.; Theera-Ampornpunt, N.; McCullough, J.S.; Speedie, S.M. The Use of Count Data Models in Biomedical Informatics Evaluation Research. *J. Am. Med. Inform. Assoc.* **2012**, *19*, 39–44. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.