# A Distributed Semantically-Enhanced Lexical Chain Text Representation Model Construction Method (Postprint)

**Authors:** Qiu Yunpeng, Wang Wenling

**Date:** 2017-11-08T00:00:00+00:00

## Abstract

## Abstract

**Purpose:** To leverage distributed semantic association for calculating lexical cohesion relations, addressing issues such as insufficient depth in inter-word relationship detection inherent in current lexical chain construction, thereby improving the quality of lexical chain construction.

**Method:** We systematically review the technical methodologies for lexical chain construction, employ WordNet lexical relations to compute semantic associations among linguistic units within texts, utilize a distributed memory model to calculate latent semantic relationships between linguistic units, and integrate these two types of semantic relations to implement a lexical chain-based text representation model. Furthermore, grounded in theoretical research, we conduct comparative experiments using scientific papers from the medical domain.

**Results:** In terms of textual thematic description, the lexical chains constructed by our method outperform those generated by non-greedy algorithms, while maintaining comparable computational time.

**Limitations:** The algorithm exhibits relatively high time consumption; does not comprehensively consider all aspects of lexical cohesion relations; and the method's effectiveness has only been validated for topic identification in medical domain scientific literature, necessitating further verification across additional domains.

**Conclusion:** Distributed semantic association can identify latent semantics, provides substantial assistance for constructing lexical chains using multi-word phrases, and effectively enhances lexical chain construction efficacy.

## Full Text

# A Method for Constructing Distributed Semantics-Enhanced Lexical Chain Text Representation Models

**Qu Yunpeng**[1],[2],[3], **Wang Wenling**[3]
[1] University of Chinese Academy of Sciences, Beijing 100049, China
[2] National Science Library, Chinese Academy of Sciences, Beijing 100190, China
[3] National Library of China, Beijing 100081, China

## Abstract

[**Objective**] This paper leverages distributed semantic associations to compute lexical cohesion relations, addressing the limitation of insufficient depth in detecting inter-word relationships in current lexical chain construction methods and improving the quality of lexical chain building. [**Methods**] We systematically review technical approaches to lexical chain construction, utilizing WordNet thesaurus relations to calculate semantic associations among linguistic units in texts, and employing a distributional memory model to compute latent semantic relationships between these units. These two types of semantic relations are combined to construct a lexical chain text representation model. Based on this theoretical framework, we conduct comparative experiments using scientific papers from the medical domain. [**Results**] In terms of text topic description, the lexical chains constructed by our method outperform non-greedy algorithms, with comparable computational time. [**Limitations**] The algorithm requires relatively long processing time; it does not comprehensively consider all lexical cohesion relations; and the method' s effectiveness has only been validated for topic identification in medical scientific literature, requiring further verification across more domains. [**Conclusions**] Distributed semantic associations can identify latent semantics and significantly facilitate lexical chain construction using multi-word phrases, effectively enhancing the quality of lexical chain building.

**Keywords:** WordNet; Distributional Memory; Lexical Chain; Distributional Semantics

## Introduction

Lexical chain text representation models construct representations of lexical cohesion relations within discourse, capturing rich semantic information. By creating an easily comprehensible contextual environment, lexical chains help determine the specific meanings of polysemous words in texts and provide clues for text structure and coherence, facilitating overall text comprehension. With their simple structure, lexical chain models have been widely applied in text segmentation [1], automatic summarization [2], text filtering [3], question answering [4], spelling error detection [5], and sentiment recognition [6].

Computational methods for lexical cohesion relations can be categorized into three types: dictionary-based, statistical, and graph-based approaches [7]. Dictionary-based methods use predefined semantic relations from lexical resources to calculate lexical cohesion, offering advantages of interpretability and implementability. These approaches have been most extensively applied and represent the primary method for lexical chain construction. For English texts, WordNet and Roget' s Thesaurus are predominantly used [8-9], while for Chinese texts, HowNet and Tongyici Cilin (Synonym Forest) are commonly employed [10-12]. Statistical methods analyze the tendency of words to co-occur around topics through statistical linguistic analysis, building co-occurrence knowledge bases that are then used to compute similarity for representing lexical cohesion. Primary algorithms include pole-based overlapping clustering [13], LDA methods [14], and E-index methods [15]. Graph-based approaches transform texts into graphs and utilize graph clustering methods to identify lexical chains [16].

Given the complementary nature of dictionary-based and statistical methods, researchers have begun exploring hybrid approaches. For instance, Marathe and Hirst attempted to combine distributional semantics with lexical resources for lexical chain construction, achieving promising results [17].

Following a comprehensive review of lexical chain construction methods, we identified several issues in current approaches to computing lexical cohesion relations. First, while dictionaries can detect explicit semantic associations and statistical information can reveal latent associations—both important types of lexical cohesion—the statistical information currently employed is relatively limited, preventing deeper exploration of potential relationships between candidate words. Second, contextual information significantly influences the computation of candidate word meanings and inter-word relations, yet current utilization of candidate word contexts remains limited. Third, although some studies have attempted to integrate dictionaries and statistical information, they have not resolved the issue of words or phrases not included in dictionaries being unable to participate in lexical chain construction.

To address these limitations, we propose a distributed semantics-enhanced lexical chain construction algorithm. Our approach employs WordNet to compute semantic associations among linguistic units in texts, utilizes a distributional memory model to calculate latent semantic relationships between units, and fuses these two types of information for lexical chain construction. The characteristics of our method are: (1) it preserves richer information from the original text by simultaneously computing both semantic and distributional semantic relations between candidate words from multiple perspectives, enabling the discovery of richer semantics; (2) it considers the impact of context on term meanings by incorporating the computation of distributional semantic association strength into the lexical chain construction process, taking into account prepositional collocations, conjunction patterns, and adjective/verb usage in the candidate word' s training context, which provides important references for

word sense disambiguation and lexical cohesion identification; and (3) it enables words or phrases not included in dictionaries to participate in lexical chain construction through computation of their distributional semantic associations and co-occurrence relations, resulting in lexical chains that contain many phrases and technical terms.

## 2. Distributed Memory Model

The fundamental theory underlying Distributional Semantics Models (DSM) is the distributional hypothesis in linguistics: "words that appear in similar contexts tend to have similar meanings" [18]. Under this assumption, a word can be mapped to a vector in distributional semantic space, where dimensions correspond to contextual environments surrounding the word and values are determined by co-occurrence information with these contexts. If two words have similar vectors, they share similar meanings [19]. DSM construction involves collecting terms' contextual environments from corpora, analyzing them, and representing the linguistic environment as a multi-dimensional vector space by computing co-occurrence information between terms and documents, contextual linguistic units, or syntactic structures. This establishes term-document matrices, term-context matrices, word-pair-pattern matrices, etc., thereby creating the distributional semantic space [20]. Such spatial models can represent semantic associations between terms, compute similarity between linguistic units, and further discover latent semantic connections. Notable DSMs include Latent Semantic Analysis [21], Random Indexing [22], Dependency Vectors, and Distributional Memory [23].

Among these models, Distributional Memory offers flexibility in rule specification and triplet utilization, making it our choice for computing distributional semantic similarity between candidate words. By defining extraction rules, Distributional Memory can extract co-occurrence information from term contexts, representing it as "term-relation-term" triplets while computing weights for each triplet to form a three-dimensional tensor <term, relation, term, value>. Unlike other distributional semantic frameworks, Distributional Memory allows free specification of relations, enabling selection of syntactic relations (e.g., prepositional relations) or any other association type connecting two terms. Additionally, it can transform the three-dimensional tensor into various two-dimensional distributional matrices as needed, such as "<term1, (relation, term2)>" matrices or "<(term1, term2), relation>" matrices, thereby representing texts from different perspectives [24]. Distributional Memory has been widely applied, with distributional memory banks for English, German [25], and Croatian [26] being constructed and applied to various natural language processing tasks.

## 3. Distributed Semantics-Enhanced Lexical Chain Construction Algorithm

The main steps of our distributed semantics-enhanced lexical chain construction algorithm include: building a candidate word list, computing semantic association relations, computing distributional semantic relations, fusing these relations, and constructing lexical chains, as shown in Figure 1 [Figure 1: see original paper]. Key issues to address include constructing the distributional semantic space, computing distributional semantic relations, computing semantic associations, fusing these relations, and the lexical chain construction algorithm itself.

### 3.1 Construction of Distributional Semantic Space and Computation of Distributional Semantic Relations

Constructing the distributional semantic space first requires identifying terms and their relationships from corpora, combining them into triplets, and then computing Local Mutual Information (LMI) values to form the space. We perform part-of-speech tagging and dependency parsing on the corpus, selecting nouns of types "NN, NNS, NNP, NNPS" and binary phrases of type "Compound" from dependency parsing results as terms. From the dependency parsing results, we select four types of association rules—prepositions, conjunctions, adjectives, and verbs—as relations in triplets, extracting triplets in the form <term, dependency type, term>. The specific rules are detailed in Table 1 .

After extracting triplets, we use the Local Mutual Information (LMI) formula [28] to compute relation weights, discarding combinations with negative LMI values. The LMI calculation formula is:

$$LMI = P(x, r, y) \log \frac{P(x, r, y)}{P(x)P(r)P(y)}$$

This transforms triplets into weighted three-dimensional tensors <term, dependency type, term, LMI>. Once all triplets are converted into weighted tensors, the distributional memory space is constructed.

### 3.2 Computation of Lexical Cohesion Relations

We first preprocess target documents using the same term extraction method as for the distributional memory space, selecting nouns of types "NN, NNS, NNP, NNPS" and binary phrases of type "Compound" from dependency parsing as candidate words for lexical chain construction. Our method computes both distributional semantic associations and dictionary-based semantic associations between candidate word pairs.

**(1) Computation of Distributional Semantic Associations**

To compute distributional semantic associations between candidate words, we dynamically extract environment vectors for candidate words from the distributional semantic space. In this space, term contexts are stored as three-dimensional tensors <term1, dependency type, term2, LMI>. During extraction, we use (dependency type, term2) as dimensions of the environment vector for candidate word x, transforming the three-dimensional tensor into a two-dimensional matrix <x, (r, y)> where matrix values correspond to LMI values. Table 2 shows example second-order vectors for terms "death"and "heart failure."

We then compute the cosine similarity between the two vectors to represent the strength of latent semantic relations between candidate words, with results directly participating in lexical chain computation. The cosine similarity formula [29] is:

$$\cos(\theta) = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

**(2) Computation of Semantic Association Relations**

For semantic association computation, we use the English lexical resource WordNet [30]. Following Silber et al.' s approach [31], we consider five semantic relations: repetition, synonymy/antonymy, holonymy/meronymy, hypernymy/hyponymy, and sibling relations. Window distances are set to 1 sentence, 3 sentences, and unlimited distance, with different weights assigned to each scenario for inter-word relation computation. Weight assignment depends on the semantic relation type and window distance, as detailed in Table 3 .

### 3.3 Relation Fusion Computation

The lexical cohesion strength between two candidate words requires fusing the two relation strengths. Experiments show that weighted fusion is reasonable, using the formula:

$$Relation(w_i, w_j) = a \times Wordnet(w_i, w_j) + b \times Dist(w_i, w_j)$$

where $Wordnet(w_i, w_j)$ represents semantic association strength, $Dist(w_i, w_j)$ represents distributional semantic relation strength, and $a$ and $b$ are empirical constants.

### 3.4 Lexical Chain Construction Algorithm

The lexical cohesion strength between a candidate word and an existing lexical chain is computed as the average strength across all words in the chain:

$$Relation(w_i, Chain) = average(Relation(w_i, w_j))$$

Following Barzilay et al.' s method [32], we process candidate words sequentially in order of appearance. For each candidate word, we compute its lexical cohesion strength with all existing chains ($Relation(w_i, Chain_j)$). If the current chain set is empty or all $Relation(w_i, Chain_j)$ values are below threshold $Thres(w, C)$, we create a new chain starting with the current candidate word; otherwise, we add the candidate word to the chain with the maximum $Relation(w_i, Chain_j)$ value. The pseudocode for lexical chain construction is:

```
Candidate lexical chain sequence LC_{List}()
For each candidate word
    For each lexical chain
        Compute relation weight Relation(w, Chain)
    End for
    If (LC_{List}() is empty) or (all scores (w, C) < Thres(w, C))
        Create new lexical chain starting with current candidate word
    Else
        Add to lexical chain with maximum relation weight
    End if
End for
```

## 4. Experiments and Analysis

We use scientific papers from the medical domain as experimental data, retrieving 100 full-text English documents from ScienceDirect using "heart" and "cardiac" as keywords to build the distributional semantic space. The constructed space contains 71,023 triplets. We use Stanford CoreNLP [33] for preprocessing, including POS tagging and stopword removal, converting texts into XML format for automated processing.

In experiments, we set empirical parameters $a$ and $b$ in the relation formula to 1, as shown in equation (5), and set the threshold $Thres(w, C)$ for adding candidate words to lexical chains at 0.5.

$$Relation(w_i, w_j) = Wordnet(w_i, w_j) + Dist(w_i, w_j)$$

### 4.1 Quality Comparison via Keyword Identification Effectiveness

We evaluate lexical chain quality from the perspective of keyword extraction results. Using "heart" and "cardiac" as keywords, we randomly selected 50 abstracts from ScienceDirect. A medical expert read and annotated 3-6 keywords per abstract. Another medical expert then reviewed lexical chains built by both the non-greedy algorithm and our algorithm, performing keyword extraction based on the chain structures. Comparing extracted keywords against the expert annotations, we computed precision and recall rates, shown in Table 4 .

The results demonstrate that our algorithm' s lexical chain construction achieves better performance than the non-greedy algorithm in topic identification.

### 4.2 Analysis of Distributional Semantics Impact on Lexical Chain Construction

We randomly selected 5 samples for statistical analysis, with results shown in Table 5 . Analysis reveals our algorithm' s superiority in three aspects: quantity of semantic information discovered, word sense determination, and candidate word identification.

**(1) Distributional semantics discovers richer semantic information.** Our algorithm results show 3,225 effective WordNet relations discovered, with 14,710 distributional semantic association computations performed, yielding 9,508 positive results. There are 6,803 term pairs with distributional semantic associations but no WordNet relations, including 347 effective distributional semantic associations involving binary phrases. Strong distributional semantic associations among term pairs without WordNet relations include <baseline function, impairment>, <patient, treatment>, <correlation, difference>, and <artery, disease>, with association strengths around 0.5. Manual analysis of original texts confirms these strong associations that WordNet cannot detect. Distributional semantics' discovery of multi-word phrases and latent semantic associations significantly impacts lexical chain construction, substantially compensating for dictionary-only limitations.

**(2) Distributional semantics analyzes candidate word meanings based on context.** For example, "evolution" has two senses: biological evolution and development/progress. The non-greedy algorithm selects the second sense, grouping "evolution" with "action." Our algorithm, through distributional semantic computation, finds stronger association between "evolution"and "origin." Similar examples include "species" (meaning "biological species" rather than "model"), which is more closely related to "human"and thus grouped accordingly.

**(3) Distributional semantics helps discover more candidate words.** Our algorithm identified 632 candidate words and 82 binary phrases across 5 test samples, retaining 464 candidate words and 80 binary phrases in final effective lexical chains. The non-greedy algorithm identified only 516 candidate words, retaining 240 in final chains—significantly fewer than our approach.

### 4.3 Algorithm Time Consumption

Table 6 shows the time consumed for lexical chain construction across 5 samples. While both algorithms have consistent time complexity, our distributional semantics-enhanced algorithm requires real-time environment vector extraction and similarity computation from the distributional semantic space, substantially increasing construction time. However, we found that distributional semantic associations between two terms remain stable when the space is stable. Therefore, we stored frequently used term pair associations in a database as an index,

improving practical efficiency to levels comparable with the non-greedy algorithm.

## Conclusion

This study' s innovative contributions are twofold: (1) We propose a distributed semantics-enhanced lexical chain construction method that strengthens semantic relations between candidate words through distributional semantic associations, enabling consideration of richer textual associations and detection of deeper lexical cohesion relations, thereby improving lexical chain quality. Experiments demonstrate our method' s superiority over non-greedy algorithms, with discovered distributional semantic relations significantly impacting and enhancing chain construction. (2) We propose an application scenario for the Distributional Memory model. As a novel model lacking extensive research in China, we pioneer its use in lexical chain construction, establishing triplet extraction rules and validating its effectiveness experimentally, laying groundwork for future research.

Future work must address several issues. While distributional semantics discovers latent associations in large-scale corpora, it cannot capture domain-specific meanings and special semantic relations within individual documents. Analyzing lexical co-occurrence in target documents could complement our method. Lexical cohesion relation detection remains incomplete. Hoey' s foundational work identifies six types of lexical cohesion relations [34], while current methods detect only three. Future research could incorporate bibliometric indices like Salton and Jaccard indices to compute lexical co-occurrence relations as supplements to distributional semantics enhancement, enabling more comprehensive lexical cohesion computation.

## References

[4] Moldovan D, Novischi A. Lexical Chains for Question Answering [C]. In: Proceedings of the 19th International Conference on Computational Linguistics-Volume Stroudsburg: Association for Computational Linguistics, 2002: 1-7.

[5] St-Onge D. Detecting and Correcting Malapropisms with Lexical Chains [D]. Toronto: University of Toronto, 1995.

[6] Naveen Kumar M, Suresh R. Emotion Detection Using Lexical Chains [J]. International Journal of Computer Applications, 2012, 57(4): 1-4.

[7] Qu Yunpeng, Wang Wenling. An Overview on the Computing Method of the Lexical Chain Text Representation [J]. Knowledge Management Forum, 2016(2): 136-144.

[8] Hirst G, St-Onge D. Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms [J]. Lecture Notes in Physics, 1995, 728(9): 123-149.

[9] Morris J, Hirst G. Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text[J]. Computational Linguistics, 1991, 17(1): 21-48.

[10] Liu Ming, Wang Xiaolong, Liu Yuanchao. Research of Key-Phrase Extraction Based on Lexical Chain [J]. Chinese Journal of Computers, 2010, 33(7): 1246-1255.

[11] Hu Xuegang, Li Xinghua, Xie Fei, et al. Keyword Extraction Based on Lexical Chains for Chinese News Web Pages[J]. Pattern Recognition and Artificial Intelligence, 2010, 23(1): 45-51.

[12] Qiu Jiangnan, Luo Zhicheng, Wang Yanzhang. Research on Semantic Relatedness Based Subjects Extraction of Emergency Plans Literature [J]. Journal of the China Society for Scientific and Technical Information, 2008, 27(6): 891-896.

[13] Dias G, Santos C, Cleuziou G. Automatic Knowledge Representation Using a Graph-based Algorithm for Language-independent Lexical Chaining [C]. In: Proceedings of the Workshop on Information Extraction Beyond the Document. Stroudsburg: Association for Computational Linguistics, 2006: 36-47.

[14] Remus S, Biemann C. Three Knowledge-free Methods for Automatic Lexical Chain Extraction [C]. In: Proceedings of NAACL-HLT 2013. Stroudsburg: Association for Computational Linguistics, 2013: 989-999.

[15] Ye Chunlei, Leng Fuhai. Study on the Keyword Extraction from Roadmap Based on the Lexical Chains [J]. New Technology of Library and Information Service, 2013(1): 50-56.

[16] Medelyan O. Computing Lexical Chains with Graph Clustering [C]. In: Proceedings of the 45th Annual Meeting of the ACL: Student Research Workshop. Stroudsburg: Association for Computational Linguistics, 2007: 85-90.

[17] Marathe M, Hirst G. Lexical Chains Using Distributional Measures of Concept Distance [C]. In: Proceedings of the 11th International Conference on Computational Linguistics. 2010: 291-302.

[18] Basili R, Pennacchiotti M. Distributional Lexical Semantics: Toward Uniform Representation Paradigms for Advanced Acquisition and Processing Tasks [J]. Natural Language Engineering, 2010, 16(4): 347-358.

[19] Molino P, Basile P, Caputo A, et al. Exploiting Distributional Semantic Models in Question Answering [C]. In: Proceedings of the 2012 IEEE 6th International Conference on Semantic Computing. Washington, DC: IEEE Computer Society, 2012: 300-303.

[20] Padó S, Lapata M. Dependency-based Construction of Semantic Space Models [J]. Computational Linguistics, 2007, 33(2): 161-199.

[21] Landauer T K, Dumais S T. A Solution to Plato' s Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge [J]. Psychological Review, 1997, 104(2): 211-240.

[22] Sahlgren M. An Introduction to Random Indexing [C]. In: Proceedings of Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, Copenhagen, Denmark. 2005.

[23] Baroni M, Lenci A. One Distributional Memory, Many Semantic Spaces [C]. In: Proceedings of the Workshop on Geometrical Models of Natural Language Semantics. Stroudsburg, PA: Association for Computational Linguistics, 2009: 1-8.

[24] Baroni M, Lenci A. Distributional Memory: A General Framework for Corpus-based Semantics [J]. Computational Linguistics, 2010, 36(4): 673-721.

[25] Padó S, Utt J. A Distributional Memory for German [C]. In: Proceedings of the KONVENS 2012. 2012: 462-470.

[26] Šnajder J, Padó S, Agić Ž. Building and Evaluating a Distributional Memory for Croatian [C]. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. 2013: 784-789.

[27] De Marneffe M-C, Manning C D. Stanford Typed Dependencies Manual [EB/OL]. [2016-04-07]. http://nlp.stanford.edu/software/dependencies_{manual}.pdf.

[28] Evert S. The Statistics of Word Cooccurrences [Elektronische Ressource]: Word Pairs and Collocations [D]. Stuttgart: University of Stuttgart, 2005.

[29] Turney P D, Pantel P. From Frequency to Meaning: Vector Space Models of Semantics [J]. Journal of Artificial Intelligence Research, 2010, 37(4): 141-188.

[30] Fellbaum C, Miller G. WordNet: An Electronic Lexical Database [M]. Cambridge, MA: MIT Press, 1998.

[31] Silber H G, Mccoy K F. Efficiently Computed Lexical Chains as an Intermediate Representation for Automatic Text Summarization [J]. Computational Linguistics, 2002, 28(4): 487-496.

[32] Barzilay R, Elhadad M. Using Lexical Chains for Text Summarization [C]. In: Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization. 1997: 10-17.

[33] Manning C D, Surdeanu M, Bauer J, et al. The Stanford CoreNLP Natural Language Processing Toolkit [C]. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. 2014: 55-60.

[34] Hoey M. Patterns of Lexis in Text [M]. Oxford University Press, 1991.

**Author Contributions:** Qu Yunpeng: conceived the research idea, designed the study, conducted experiments, and drafted the manuscript; Wang Wenling: collected, cleaned, and analyzed data, and revised the manuscript.

**Conflict of Interest Statement:** All authors declare no conflict of interest.

**Supporting Data:** Supporting data [1,3] are available in the journal' s online version at http://www.infotech.ac.cn; supporting data [2] is stored by the authors and available via E-mail: quyp@nlc.cn.

[1] Qu Yunpeng. Supporting Data.xlsx. Keyword quality comparison data for Section 4.1.
[2] Qu Yunpeng. Code.zip. Source code involved in this paper.
[3] Qu Yunpeng. Corpus and Results.zip. Test corpora and generated lexical chain results.

*Note: Figure translations are in progress. See original paper for figures.*