Research on Chinese Sequence Alignment Based on Word Vector Models (Postprint)

Authors: Huixiang Xiong, Zhao Dengpeng, Lu Chenfan

Date: 2023-04-01T00:00:00+00:00

Abstract

[Purpose/Significance] This study addresses the application of renowned sequence alignment algorithms from bioinformatics to text similarity, improving upon existing methods and enhancing the accuracy of text similarity computation.

[Method/Process] First, target texts are normalized to construct a Chinese sequence set. Subsequently, a trained Skip-Gram model from Word2vec is employed to build a word-pair scoring matrix for this Chinese sequence set and to establish scoring rules. Finally, pairwise global alignment is performed on the Chinese sequences to obtain the optimal alignment solution; by backtracking to obtain the alignment path of this optimal solution, the similarity between Chinese sequences is calculated.

[Results/Conclusion] Empirical results demonstrate that compared with traditional methods, the proposed method, which integrates word vector models, improves the accuracy of text similarity computation and effectively resolves the issue of duplicate word pairs that occurs in traditional methods.

Full Text

Preamble

Volume 64, Issue 10, May 2020

Research on Chinese Sequence Alignment Based on Word Vector Models

Xiong Huixiang[1], Zhao Dengpeng[1], Lu Chenfan[2]
[1]School of Information Management, Central China Normal University, Wuhan 430079
[2]School of Statistics and Management, Shanghai University of Finance and Economics, Shanghai 200433

Abstract:
[Purpose/Significance] This study addresses the application of renowned sequence alignment algorithms from bioinformatics to text similarity computation, improving upon previous methods to enhance the accuracy of text similarity calculation. [Method/Process] First, target texts undergo normalization processing to form a Chinese sequence set. Subsequently, the trained Skip-Gram model from Word2vec is employed to construct a word-pair scoring matrix for this Chinese sequence set, with scoring rules established. Finally, global alignment is performed pairwise on Chinese sequences to obtain the optimal alignment solution, the alignment path of this optimal solution is backtracked, and the similarity between Chinese sequences is calculated. [Result/Conclusion] Empirical results demonstrate that compared with traditional methods, the proposed approach integrates word vector models to improve the accuracy of text similarity computation and effectively resolves the problem of repeated word pairs encountered in conventional methods.

Keywords: Word2vec; Chinese sequence; sequence alignment; global alignment; text similarity
Classification Number: TP391.1
DOI: 10.13266/j.issn.0252-3116.2020.10.010

Introduction

Text similarity computation refers to the process of comparing two or more entities (words, short texts, documents) using specific strategies to obtain a quantified numerical value representing their similarity. With the rapid development of information technology, mining and analyzing the massive amounts of information generated on the Internet can provide users with relevant and meaningful content, such as personalized recommendations and intelligent retrieval. As a bridge connecting fundamental research and upper-layer applications, text similarity algorithms have emerged in numerous text mining domains including natural language processing, text classification, text clustering, question-answering systems, information retrieval, and search engines, gaining extremely widespread application [1]. Currently, an increasing number of text similarity computation methods exist. Wang Chunliu et al. [2] reviewed classic literature in the text similarity computation field over the past two decades, elaborating on both surface text similarity and semantic similarity computation methods, with corpus-based approaches representing the primary research direction in semantic similarity. Most scholars recognize classification schemes based on string methods, corpus-based methods, knowledge-based methods, and hybrid approaches [3-5]. Previous studies on text similarity algorithms have included string-based methods such as edit distance [6], longest common subsequence [7], Hamming distance [8], and N-gram models [9], among which sequence alignment algorithms belong to the longest common subsequence method and demonstrate good performance for streaming data and time-series data [10]. In Chinese information processing, computing similarity between Chinese character strings, such

as words and phrases, plays a crucial role in dictionary compilation, example-based machine translation, automatic question-answering, and information filtering [11]. Furthermore, depending on the granularity of compared characters and alignment methods, sequence alignment algorithms can also be applied to semantic mining, text classification and clustering, personalized recommendations, and intelligent retrieval.

Sequence alignment algorithms originate from the field of bioinformatics, representing the most commonly used and classic research method for analyzing sequences to understand gene structure and function. Typically, these algorithms perform pairwise comparisons between protein or nucleic acid sequences, identifying similar regions and conserved sites to seek homologous structures and reveal issues of biological evolution, heredity, and variation [12]. Based on the number of sequences aligned simultaneously, sequence alignment algorithms are classified as pairwise or multiple sequence alignment. In 1970, S.B. Needleman and C.D. Wunsch proposed the global pairwise sequence alignment algorithm [13]. In 1975, T.F. Smith and M.S. Waterman improved upon this with a local pairwise alignment algorithm [14]. As the number and length of aligned sequences increased, D.F. Feng and R.F. Doolittle introduced multiple sequence alignment algorithms in 1987 [15]. Subsequently, with the continuous development of bioinformatics, numerous sequence alignment tools and software emerged and were refined, including BLAST [16], HMM [17], CLUSTALW [18], and T-COFFEE [19]. Recent research on sequence alignment algorithms has focused primarily on improvements and acceleration of multiple sequence alignment to facilitate deeper study of genes and proteins [20-21].

In 2010, Xu Shuo [22] proposed a novel method for Chinese term semantic similarity computation based on pairwise alignment, identifying and overcoming issues in traditional semantic similarity methods but without considering the impact of word order on similarity computation in special cases—such as when adjacent words in compared texts are swapped without changing meaning (e.g., <anxiety, depression> vs. <depression, anxiety>), where the method yields a similarity of 0.5 when it should be 1. In Wang Ting's [23] global alignment algorithm, referencing Tian Jiule's [24] use of Tongyici Cilin (a Chinese thesaurus) to compute word similarity improved the accuracy of sequence alignment for text similarity computation, but this approach only works well for texts whose words are all contained in the thesaurus and fails to consider inter-word relationships.

Word2vec and Sequence Alignment Algorithms

Word2vec

Word2vec is a word vector model developed by Google in 2013 based on deep learning concepts, primarily used to transform unstructured text information into vectorized form [26]. Since its release, Word2vec has been widely applied in natural language processing, with related research gradually increasing, making it one of the most representative tools in the field [27]. Word2vec can convert

words into vector representations through text learning, using word vectors to represent semantic information [27]. As a natural language processing tool, its greatest feature is context-based word representation, solving the "curse of dimensionality" problem. Based on different training methods, Word2vec can be divided into CBOW and Skip-Gram models. CBOW uses a word's context as input to predict the word itself, while Skip-Gram uses the word as input to predict its context. Comparatively, the CBOW model performs better on small corpora, whereas the Skip-Gram model is more suitable for large corpora [26, 28].

Sequence Alignment Algorithms

Sequence alignment algorithms, originating from bioinformatics, typically involve arranging two DNA or amino acid sequences side-by-side to identify similarities, allowing for gap insertion so that as many identical or similar symbols as possible align in the same columns. Broadly, these algorithms fall into two categories: one proposed by S.B. Needleman and C.D. Wunsch [13] for comparing overall sequence similarity, known as global alignment; the other by T.F. Smith and M.S. Waterman [14] for comparing partial sequence fragments, known as local alignment [29]. Current applications of sequence alignment algorithms to text similarity in library and information science primarily focus on global alignment, which treats text words as characters for comparison, aligning identical words in the same columns and scoring them according to predefined rules to compute text similarity. With bioinformatics development, to achieve higher-quality alignment results, sequence alignment algorithms reference scoring matrices derived from probability statistics of numerous nucleic acids (DNA and RNA) or amino acids.

BLOSUM62 Scoring Matrix In 1992, S. Henikoff and J.G. Henikoff addressed distant sequence relationships by deriving substitution matrices BLOSUM from protein block databases and presented the design concept and methodology for the BLOSUM62 matrix [30-31]. The BLOSUM62 matrix employs log-odds scoring, where the log-odds value is the natural logarithm of the ratio of homologous to non-homologous likelihoods. For two residues a and b (treated as amino acids), the score s(a,b) in BLOSUM62 is calculated as:

$$s(a, b) = \text{formula (1)}$$

where f_a and f_b represent the average background frequencies of residues a and b assuming they are non-homologous and independent; P_{ab} is the target frequency of residues a and b appearing as homologous in known homologous sequences; and λ is a scaling parameter. Using background frequencies of various amino acid residues and formula (1), a BLOSUM62 scoring matrix is obtained, as shown in [Figure 1: see original paper].

Dynamic Programming Algorithm: Needleman-Wunsch In 1970, S.B. Needleman and C.D. Wunsch proposed the global alignment algorithm [13], which analyzes the relationship between two sequences holistically—considering total sequence length to maximize global similarity. This method belongs to the dynamic programming algorithm family [32], applied to optimization problem solving by combining subproblem solutions to achieve an overall optimal solution. In sequence alignment, the dynamic programming concept states that any point terminating on an optimal path must itself lie on the optimal subpath terminating at that point [33].

The global alignment algorithm's basic principle involves placing two sequences to be aligned in a two-dimensional table along horizontal and vertical axes. Alignment begins from the Gap position. During alignment, each position has three extension possibilities: (1) diagonal extension—if characters match, a reward score from the scoring matrix is given; if mismatched, a penalty is applied; (2) vertical extension—when the vertical sequence's character cannot match the horizontal sequence's corresponding character, a gap symbol "-" is inserted in the horizontal sequence to match the vertical sequence's character, incurring a penalty; (3) horizontal extension—conversely, when the horizontal sequence's character cannot match, a gap is inserted in the vertical sequence. The "-" symbol indicates that to achieve an overall optimal alignment result, one sequence may correspond to a "-" at certain positions. Continuous iterative extension yields multiple final solutions, each being the sum of all reward and penalty scores during alignment. The highest score is taken as the optimal solution, and backtracking reveals the alignment path, producing two final sequences of equal length. Backtracking from the bottom-right corner according to the optimal solution's score yields the complete alignment path. However, the global alignment algorithm is recursive, involving many repeated calculations—obtaining the optimal score and path requires operations proportional to the matrix size n×m, with time complexity $O(n^2)$. Backtracking ensures alignment completeness and accuracy while obtaining sequence alignment length, as gap insertion changes the original sequence length.

[Figure 2: see original paper] illustrates a partial alignment process between Sequence 1 = {V, D, S, C, Y} and Sequence 2 = {V, E, S, L, C, Y}. Referencing the scoring matrix in [Figure 1: see original paper] and defining scoring rules—matching receives a reward score (from [Figure 1: see original paper]), while mismatches or gaps receive a penalty Gap = -11 (commonly used in bioinformatics research, though adjustable for better results). Alignment begins from Gap, where gap-gap correspondence scores 0. From Gap, continuous alignment extension yields three possibilities at each position. For example, diagonal alignment matches character "V" from both sequences, receiving 4 points for exact match. Horizontal alignment cannot match sequence characters, so Sequence 2 inserts a "-" to match Sequence 1's "V", penalizing 11 points (-11), called a gap penalty. Conversely, vertical alignment matches Sequence 2's "V" with Sequence 1's "-", also penalizing 11 points. In [Figure 2: see original paper], Sequence 1's second character "D" and Sequence 2's second character "E" differ,

yet still receive 2 points from the scoring matrix because under bioinformatics amino acid statistics, these characters have certain similarity—representing similar matching, analogous to aligning semantically similar words like "happy" and "joyful". Adding the previous position's "V"-"V" score of 4 yields 6 points at the current position. Through continuous recursive extension, the solution with the highest cumulative score is taken as optimal, and backtracking its alignment path yields the result shown in .

Improved Chinese Sequence Alignment Algorithm

Building upon previous research, this paper proposes an improved Chinese sequence alignment algorithm that applies Word2vec trained on Baidu Baike corpus to global alignment. First, referencing existing literature [22-23], we define: words existing in the corpus as *basic words*; words appearing in the empirical dataset but not in the corpus as *non-basic words*; and for two texts after word segmentation, if any two adjacent words in one text match any two adjacent words in another text after swapping order, these adjacent words are merged and called *repeated words*, forming *repeated word pairs*. For example, if adjacent words {anxiety, depression} and {depression, anxiety} appear in two texts to be compared, they are merged into repeated words {anxiety-depression} and {depression-anxiety}, constituting a repeated word pair. The Chinese sequences compared in this paper comprise basic words, non-basic words, repeated words, and other textual sequences.

Given a text collection, after a series of normalization processes, we obtain a Chinese sequence set: CS = {cs_1, cs_2, …, cs , …, cs }, where cs = {t ,$_1$, t ,$_2$, …, t , , …, t , }, and t , represents the k-th word in the i-th Chinese sequence. First, Word2vec represents sequence words as word vectors, calculating cosine similarity between t , and t , to construct a word-pair scoring matrix and define scoring rules. Then for any two Chinese sequences cs = {t ,$_1$, t ,$_2$, …, t , , …, t , } and cs = {t ,$_1$, t ,$_2$, …, t , , …, t , } in CS, sequence alignment obtains the optimal solution, backtracks the alignment path, and finally calculates sequence similarity sim(cs , cs).

Traditional Sequence Alignment Method

Traditional sequence alignment methods treat segmented text words as characters for comparison. As shown in [Figure 3: see original paper], target texts are first preprocessed into a normalized Chinese sequence set. For any two Chinese sequences cs and cs , global alignment is performed after defining scoring rules (formula (2)): if compared words are identical or similar, they receive a score sim(t , , t ,) between 0-1.0. To obtain global optimal solutions, gap penalties exist at certain positions, with Gap = -0.05 (following previous research). After aligning cs and cs to obtain the optimal solution and backtracking its path, similarity is calculated using formula (3).

Formula (2):

sim(t , , t ,) if t , ≠ "-", t , ≠ "-"
sim(t , , -) = Gap = -0.05 if t , = "-"
sim(-, t ,) = Gap = -0.05 if t , = "-"

Formula (3):
sim(cs , cs) = Σ sim(t , , t ,)

This transforms Chinese sequence alignment into a dynamic programming problem using the Needleman-Wunsch global alignment algorithm to recursively find optimal solutions. Taking bipolar depression anxiety and bipolar anxiety depression as examples, after word segmentation we obtain cs = {bipolar, affective, depression, anxiety} and cs = {bipolar, affective, anxiety, depression}. The alignment process is shown in [Figure 4: see original paper]. Due to Chinese language's general "light front, heavy back" structure, alignment proceeds from end to beginning. Starting from GAP, vertical arrows indicate vertical extension (inserting a gap in cs), horizontal arrows indicate horizontal extension (inserting a gap in cs), ensuring final sequences of equal length with alignment length L = 5.

In [Figure 4: see original paper], the solution with final score 2.90 is optimal. Backtracking this optimal alignment path yields two sequences of equal length shown in . Backtracking ensures correct optimal alignment results while determining final sequence length, as gap insertion changes original sequence length. Finally, applying formula (3), the similarity between the two Chinese sequences is sim(cs , cs) = (1 + 1 - 0.05 + 1 - 0.05)/5 = 0.58.

Word Similarity Computation Based on Word2vec

The core of this method is constructing a word-pair scoring matrix for Chinese sequence alignment reference, where the matrix essentially contains vector cosine values between words in any two sequences cs and cs . Word2vec operates on the principle that words with similar contexts have approximate meanings. After extensive corpus training, it effectively represents word vectors and quantifies word-pair relationships through cosine similarity calculation—closely resembling bioinformatics' approach of building nucleic acid and protein scoring matrices through extensive statistics. Therefore, this paper uses the comprehensive Baidu Baike corpus to train Word2vec's Skip-Gram model and compute Chinese sequence word-pair similarity [34].

Word similarity computation is illustrated in [Figure 5: see original paper]. Target texts are preprocessed and segmented into normalized CS. Rather than randomly comparing Chinese sequences in CS pairwise, complete CS is first divided into two Chinese sequence sets for comparison. For any cs and cs in these sets, alignment is performed after constructing the scoring matrix.

Before alignment, we must first obtain word similarities to build the scoring matrix. The word similarity set computation process is as follows:

1. Extract all words from the two Chinese sequence sets to be compared,

cs = {t ,1, t ,2, ..., t , , ..., t , } and cs = {t ,1, t ,2, ..., t , , ..., t , }, and deduplicate to avoid extensive repeated computation.

2. Remove from the word set any special terms or English abbreviations in cs and cs that are not contained in the trained Word2vec model.
3. Vectorize remaining words t , and t , present in the model using trained Word2vec and calculate cosine similarity to construct the word-pair scoring matrix.
4. Traverse all cs and cs to identify repeated word pairs: for any adjacent words t , and t , 1 in cs and t , and t , 1 in cs , swap adjacent word positions in cs and merge them into single-word forms t , 1t , and t , t , 1 for mutual matching. Continuously form word groups from adjacent words in both sequences and match them until the last word groups are matched, filtering out exact matches as repeated word pairs. Using reference [24]'s similarity computation method and word vector model, calculate repeated word pair similarity as sim(t , t , 1, t , t , 1); if uncomputable, default to sim(t , t , 1, t , t , 1) = 1.

Improved Chinese Sequence Alignment Algorithm

Traditional methods treat Chinese sequence words as characters for alignment, ignoring semantic meaning. When compared Chinese sequences contain repeated word pairs, effectiveness is compromised. This paper proposes a Chinese sequence alignment method based on word vector models, using Word2vec to construct word-pair scoring matrices that consider inter-word semantics and improve alignment effectiveness. The established rules also address repeated word issues in compared Chinese sequences.

Key concepts are declared as follows:

Scoring threshold λ (0-1.0): word pairs with vector cosine values exceeding this value enter the scoring matrix; others go to the non-scoring lexicon.

Word-pair scoring matrix: contains all word pairs with cosine similarity $>$ λ for Chinese sequence alignment reference.

Non-scoring lexicon: contains word pairs with cosine similarity $<$ λ and word pairs whose similarity cannot be computed via Word2vec; if λ changes, the non-scoring lexicon adjusts the scoring matrix.

Scoring rules: during alignment, if both compared words are in the scoring matrix, score according to matrix values; gap penalties are -0.05; mismatches are -0.05.

The specific method is shown in [Figure 6: see original paper]. First, target texts from online health information and academic websites are preprocessed, segmented, and represented as normalized CS. For any two sequences cs and cs to be compared, determine if words t , and t , are in the corpus. If not, place them in the "non-scoring lexicon"; if yes, represent t , and t , as spatial feature vectors via Word2vec and compute cosine similarity sim(t , , t ,). If sim(t , , t ,) $>$ λ, place the word pair in the scoring matrix; otherwise, place it in the non-scoring lexicon.

Before alignment, traverse any two Chinese sequences cs and cs to identify those containing repeated word pairs, placing them in "to-be-processed CS". Sequences without repeated word pairs go directly to "normalized CS". Then, place all identified repeated word pairs into the segmentation dictionary and re-segment "to-be-processed CS" to form "normalized CS". Using cs = {data, based on, hybrid, collaborative, filtering, dynamic, user, personalized, recommendation} and cs = {based on, data, community, personalized, recommendation, system} as examples, traversing reveals the repeated word pair {data-based on, based on-data}. Since {data, based on} and {based on, data} are initially separate, placing {data-based on, based on-data} into the segmentation dictionary and re-segmenting yields cs = {data-based on, hybrid, collaborative, filtering, dynamic, user, personalized, recommendation} and cs = {based on-data, community, personalized, recommendation, system}, now in "normalized CS". Using reference [24]'s method and word vector model, compute repeated word pair similarity; if uncomputable, default similarity is 1. Compare with λ: if $< \lambda$, place in non-scoring lexicon; if $> \lambda$, place in scoring matrix. Finally, align sequences in "normalized CS" using the scoring matrix and rules, obtain the optimal solution, backtrack its path, and compute similarity using formula (3).

Empirical Study

This section demonstrates the algorithm's applicability using two data types.

Chinese Sequence Acquisition and Normalization

Empirical data were selected from online health information websites and online academic websites. Online health consultation texts are relatively complex with widely distributed user groups, containing many non-standard expressions, typos, and missing words. Online academic website data mostly follows strict paradigms with standardized expressions. After comprehensive evaluation of domestic online health information website rankings based on Baidu weight, PR values, traffic rankings, and considerations of reliability, completeness, and accessibility, "Haodf.com" was selected as the target health information website. A crawler program designed with PyCharm collected 5,658 patient-doctor consultation texts from January 1, 2019 to February 27, 2019 as the online health information dataset. Among numerous academic websites, CNKI was chosen for its intuitive interface, simple retrieval, and comprehensive paper coverage. From CNKI, 753 Chinese literature titles retrieved with "personalized recommendation" as keyword from February 2017 to February 2019 were exported as the online academic resource dataset.

For the online health information dataset, invalid data such as "Help me!", "Urgent!", and "Post-diagnosis report" were first removed. Since some patients only posted questions without follow-up consultations, data with $ $2 doctor-patient exchanges were eliminated, resulting in 5,480 valid patient consultation texts.

The first dialogue from each patient was selected as the target sequence (typically containing comprehensive self-descriptions of condition), yielding 5,480 online health information Chinese sequences after segmentation. For the academic resource dataset, though already standardized, some titles retrieved with "personalized recommendation" were unrelated to the topic despite content relevance; these were retained. All titles were segmented using Python's jieba, producing 753 online academic resource Chinese sequences. Sample data are shown in .

Word-Pair Scoring Matrix Construction

For Word2vec training, using empirical data as corpus was considered, but online health information contains non-standard words and segmentation issues with unknown words, making the trained model non-generalizable and limited by small data volume. Since the data consist mainly of patients' condition descriptions with minimal medical terminology, and considering Word2vec's requirement for comprehensive, large-scale corpora, the Baidu Baike corpus was selected for training. Trained Word2vec represented all words present in Baidu Baike from the Chinese sequence set as spatial feature vectors, computing inter-word cosine similarities shown in and .

Setting $\lambda = 0.5$, word pairs with vector cosine values $> \lambda$ enter the scoring matrix; others enter the "non-scoring lexicon," yielding word-pair scoring matrices for both online health information and academic resource Chinese sequences. Boxed values correspond to word pairs meeting $\lambda = 0.5$.

Before Chinese sequence alignment, traverse any two sequences cs and cs to identify those containing repeated word pairs, placing them in "to-be-processed CS". Sequences without repeated word pairs go to "normalized CS". Then, place all repeated word pairs into the segmentation dictionary, re-segment all sequences in "to-be-processed CS," and incorporate them into "normalized CS". Using cs = {data, based on, hybrid, collaborative, filtering, dynamic, user, personalized, recommendation} and cs = {based on, data, community, personalized, recommendation, system} as examples, traversal reveals the repeated word pair {data-based on, based on-data}. Initially separate as {data, based on} and {based on, data}, placing {data-based on, based on-data} into the segmentation dictionary and re-segmenting yields cs = {data-based on, hybrid, collaborative, filtering, dynamic, user, personalized, recommendation} and cs = {based on-data, community, personalized, recommendation, system} in "normalized CS". Compute repeated word pair similarity using reference [24]'s method and word vector model; if uncomputable, default similarity is 1. Compare with λ: if $< \lambda$, place in non-scoring lexicon; if $> \lambda$, place in scoring matrix. Now with complete "normalized CS" and word-pair scoring matrix, align sequences using the scoring matrix and rules.

Chinese Sequence Alignment

After the previous steps, we obtain "normalized CS" and the constructed word-pair scoring matrix. Due to Chinese grammar complexity and flexible expression, Chinese generally exhibits a "light front, heavy back" pattern, so alignment proceeds from end to beginning. The optimal alignment solution is obtained, its path backtracked, and similarity between Chinese sequences calculated. This paper denotes online health information Chinese sequences as OHICS and academic resource sequences as OARCS, demonstrating partial alignment results and optimal paths.

For compared cs and cs , multiple solutions exist during alignment. The algorithm designed with Pycharm directly recursively identifies the optimal solution among many and backtracks its path. Sample sequences are shown in (OHICS) and (OARCS), where left sequences are compared with right sequences. For OHICS, sequences with IDs cs_1 and cs_2 are compared, cs_3 and cs_4 are compared, and so on until all cs and cs pairs are processed.

Alignment of sequences in and yields optimal paths shown in and ($\lambda = 0.5$). Backtracking the optimal path essentially obtains scoring and penalty results for each position, ensuring final sequence equality, then computes similarity using all scores, penalties, and sequence length.

Experimental Results and Evaluation

To demonstrate differences and superiority over traditional methods, this paper adjusts the word-pair scoring matrix during Chinese sequence alignment: (1) Traditional method: leave scoring matrix empty without reference; (2) Proposed method ($\lambda = 0.5$): include word pairs with similarity > 0.5; (3) Proposed method ($\lambda = 0$): include word pairs with similarity > 0. After alignment, similarities are computed under different conditions.

Comparing a set of OHICS and OARCS optimal alignment paths and scores (shown in and) reveals that traditional methods lack a reference scoring matrix, causing many meaningful and related word pairs to remain unaligned and producing more gap penalties, resulting in poorer alignment quality. In contrast, the proposed method using Word2vec to construct the scoring matrix aligns more relevant, meaningful word pairs and reduces gap penalties. Adjusting λ further optimizes alignment by identifying more meaningful, interrelated word pairs.

Similarity computation results for OHICS and OARCS using traditional and proposed methods are shown in and . The proposed method clearly improves Chinese sequence similarity computation by considering word semantics and relationships, though some results remain identical to traditional methods when no reference word pairs exist in the scoring matrix.

Plotting and as line charts in [Figure 7: see original paper] (x-axis: ten Chinese sequence groups; y-axis: similarity scores) visually demonstrates overall

improvement across both datasets, with more pronounced effects in health information sequences. Deduplication of word pairs reveals that for the ten OHICS groups, the scoring matrix contains 253 qualifying word pairs when $\lambda = 0.5$, and 1,581 when $\lambda = 0$. For the ten OARCS groups, only 224 word pairs qualify when $\lambda = 0.5$, and 1,616 when $\lambda = 0$.

Comparing these word pairs shows that academic paper titles use more academic, objective, and professional terminology with higher content discrimination. Health information texts, being patient self-descriptions of conditions, contain many similar symptom features across different patients, making inter-word connections more prevalent. Consequently, similarity computation performs better for OHICS than OARCS.

Addressing traditional methods' failure to handle repeated word pairs, demonstrates that in cs and cs alignment, only personalized or mobile can exactly match, with other positions incurring mismatch penalties or gaps, increasing penalties or sequence length and yielding lower similarity. The proposed method identifies and merges repeated words before alignment, revealing more similarities between Chinese sequences.

Comparing optimal alignment paths, both methods produce final length 8 sequences. Traditional alignment only matches personalized-personalized for 1 point, with gap penalties of -0.05 at first position and mismatch penalties of -0.05 elsewhere, yielding sim(cs , cs) = (1 + (-0.05)$×7)/8 = 0.08125.*Theproposedmethod*($ = 0.5) yields sim(cs , cs) = (1 + 0.54 + 0.65 + (-0.05)$×$5)/8 = 0.2425; with $\lambda = 0$, sim(cs , cs) = (1 + 0.097 + 0.339 + 0.544 + 0.651 + (-0.05)$×$3)/8 = 0.310.

Traditional sequence alignment strictly follows word order, causing significant errors when repeated words exist. The proposed method's handling of repeated word pairs improves accuracy. The main difference lies in scoring rule processing and scoring matrix construction. Traditional methods strictly compare words, ignoring semantics, and typically align only exact matches that don't necessarily reflect true text similarity. The proposed method based on Word2vec aligns both semantically related general words and significantly increases alignment of professional terms, providing complete alignment paths for subsequent research.

When using trained Word2vec to compute cosine similarity, although it effectively represents training corpus words as feature vectors, these vectors cannot reflect word order in text. Due to Chinese grammar complexity and flexible expression, word order is crucial for content representation. The proposed global alignment algorithm strictly follows word order while using Word2vec's scoring matrix, enhancing method effectiveness and demonstrating its advantages.

Limitations and Future Work

Despite proposing an improved sequence alignment algorithm for text similarity computation, several limitations remain: First, the corpus selected for Word2vec

training is central to scoring matrix construction; without good specificity and coverage, effectiveness is impacted. Second, repeated word pair similarity computation is crucial, but when corpora, thesaurus dictionaries, or previous similarity methods cannot compute repeated word pair similarity, objective and reasonable results cannot be obtained, reducing accuracy. Third, coordinating conjunctions (he, yu, qie, etc.) expressing parallel relationships are often order-independent and may affect accuracy, but diverse parallel expression forms require further research to improve the algorithm.

In bioinformatics, sequence alignment algorithms are primarily used for phylogenetic tree construction (sequence classification and clustering) and identifying sequence similarities for further research. Comparing bioinformatics sequence alignment with Chinese text research reveals many commonalities. Future work will reference more mature bioinformatics methods combined with library and information science research to attempt constructing Chinese text "phylogenetic trees" and conduct deeper, more meaningful research on Chinese semantics and similarity.

References

[1] Zhang Jinpeng. Research and Application of Semantic-Based Text Similarity Algorithms[D]. Chongqing: Chongqing University of Technology, 2014.
[2] Wang Chunliu, Yang Yonghui, Deng Fei, et al. Survey of Text Similarity Computation Methods[J]. Information Science, 2019, 37(3): 158-168.
[3] Gomaa WH, Fahmy AA. Short Answer Grading Using String Similarity and Corpus-Based Similarity[J]. International Journal of Computer Applications, 2013, 68(13): 13-18.
[4] Kadupitiya J, Ranathunga S, Dias G. Short Sentence Similarity Calculation Using Corpus-Based and Knowledge-Based Similarity Measures[C]//Proceedings of the 26th International Conference on Computational Linguistics. Osaka: The COLING 2016 Organization Committee, 2016: 44-53.
[5] Gomaa WH, Fahmy AA. A Survey of Text Similarity Approaches[J]. International Journal of Computer Applications, 2013, 68(13): 13-18.
[6] Levenshtein VI. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals[J]. Soviet Physics Doklady, 1966, 10(8): 707-710.
[7] Melamed ID. Automatic Evaluation and Uniform Filter Cascades for Inducing N-Best Translation Lexicons[EB/OL]. [2019-07-05]. https://arxiv.org/abs/cmp-lg/9505044.
[8] Zhang Huanjiong, Wang Guosheng, Zhong Yixin. Text Similarity Computation Based on Hamming Distance[J]. Computer Engineering and Applications, 2001(19): 21-22.
[9] Kondrak G. N-Gram Similarity and Distance[EB/OL]. [2019-07-05]. https://link.springer.com/chapter/10.1007%2F11575832_{13}#citeas.
[10] Bobadilla J, Ortega F, Hernando A, et al. Improving Collaborative Filtering Recommender System Results and Performance Using Genetic Algorithms[J]. Knowledge-Based Systems, 2011, 24(8): 1310-1316.

[11] Zhang Chengshi. A Multi-Layer Feature-Based Chinese String Similarity Computation Model[J]. Journal of Intelligence, 2005, 24(6): 696-701.
[12] Wen Fengchun, Wang Bangju, Xiao Zhihong. Research Status of Biological Sequence Alignment Algorithms[J]. Bioinformatics, 2010, 8(1): 66-69.
[13] Needleman SB, Wunsch CD. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins[J]. Journal of Molecular Biology, 1970, 48(3): 443-453.
[14] Smith TF, Waterman MS, Fitch WM. Comparative Biosequence Metrics[J]. Journal of Molecular Evolution, 1981, 18(1): 38-46.
[15] Feng DF, Doolittle RF. Progressive Sequence Alignment as a Prerequisite to Correct Phylogenetic Trees[J]. Journal of Molecular Evolution, 1987, 25(4): 351-360.
[16] Altschul SF, Gish W, Miller W, et al. Basic Local Alignment Search Tool[J]. Journal of Molecular Biology, 1990, 215(3): 403-410.
[17] Eddy SR. Multiple Alignment Using Hidden Markov Models[J]. International Conference on Intelligent Systems for Molecular Biology, 1995(3): 114-120.
[18] Thompson JD, Higgins DG, Gibson TJ. Clustal W: Improving the Sensitivity of Progressive Multiple Sequence Alignment Through Sequence Weighting, Position-Specific Gap Penalties and Weight Matrix Choice[J]. Nucleic Acids Research, 1994, (22)22: 4673-4680.
[19] Notredame C, Higgins J. T-Coffee: A Novel Method for Fast and Accurate Multiple Sequence Alignment[J]. Journal of Molecular Biology, 2000, 302(1): 0-217.
[20] Lassmann T. Kalign3: Multiple Sequence Alignment of Large Datasets[J]. Bioinformatics, 2019, 36(6): 1928-1929.
[21] Zhang CX, Zheng W, Mortuza SM, et al. DeepMSA: Constructing Deep Multiple Sequence Alignment to Improve Contact Prediction and Fold Recognition for Distant Homology Proteins[J]. Bioinformatics, 2019, 36(7): 2105-2112.
[22] Xu Shuo, Zhu Lijun, Qiao Xiaodong, et al. A New Method for Chinese Term Semantic Similarity Computation Based on Pairwise Alignment[J]. Journal of Intelligence, 2010, 29(4): 701-708.
[23] Wang Ting, Xu Tiansheng, Ji Fujun. Large-Scale Chinese Linked Data Model Based on Data Fields and Global Sequence Alignment[J]. Chinese Journal of Information Processing, 2016, 30(3): 204-212.
[24] Tian Jiule, Zhao Wei. Word Similarity Computation Method Based on Tongyici Cilin[J]. Journal of Jilin University (Information Science Edition), 2010, 28(6): 602-608.
[25] Tang Xiaobo. Semantic Annotation of Text Knowledge Fragments Based on Ontology and Word2Vec[J]. Information Science, 2019, 37(4): 97-102.
[26] Mikolov T, Sutskever I, Chen K, et al. Distributed Representations of Words and Phrases and Their Compositionality[J]. Advances in Neural Information Processing Systems, 2013(26): 3111-3119.
[27] Guo Sicheng, Li Gang, Zhou Huayang. Research on Medical Knowledge Organization System Interoperability Based on Word2Vec—Taking Semantic Mapping Between Vocabularies as an Example[EB/OL]. [2019-08-17].

https://kns.cnki.net/KCMS/detail/11.1762.G3.20190528.1151.002.html.
[28] Xu CZ, Liu D. Chinese Text Summarization Algorithm Based on Word2vec[EB/OL]. [2019-07-05]. https://iopscience.iop.org/article/10.1088/1742-6596/976/1/012006.
[29] Stephen FA, Thomas L, Alejandro A, et al. Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs[J]. Nucleic Acids Research, 1997, 25(17): 3389-3402.
[30] Henikoff S, Henikoff JG. Amino Acid Substitution Matrices from Protein Blocks[J]. Proceedings of the National Academy of Sciences of the United States of America, 1992, 89(22): 10915-10919.
[31] Eddy SR. Where Did the BLOSUM62 Alignment Score Matrix Come From?[J]. Nature Biotechnology, 2004, 22(8): 1035-1036.
[32] Sankoff D. The Early Introduction of Dynamic Programming into Computational Biology[J]. Bioinformatics, 2000, 16(1): 41-47.
[33] Zhang Fuxiang, Zhou Jinling. Parallelization Research and Application of Sequence Alignment Algorithms[J]. Journal of Weifang University, 2008, 8(4): 85-87.
[34] Zhao Dengpeng. Word2vec Training Corpus[EB/OL]. [2019-05-17]. https://pan.baidu.com/s/1TZ8GII0CEX32ydjsfMc0zw#list/path=%2F.

Author Contributions

Xiong Huixiang: Proposed research direction and methodology
Zhao Dengpeng: Data acquisition and paper writing
Lu Chenfan: Technical support and guidance

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