A Cross-Language Plagiarism Detection Technique Based on Fingerprint Fusion (Postprint)

Authors: Liu Gang, Zuo Quan, Yang Qianru

Date: 2018-05-20T00:00:00+00:00

Abstract

Cross-language plagiarism has consistently been a major hotspot for academic misconduct and an exceptionally difficult-to-detect form of plagiarism. The detection and identification technology for cross-language plagiarism represents the most urgently needed development and the greatest technical challenge in the field of anti-plagiarism. Based on a comprehensive summary and analysis of the domestic and international research status of both monolingual and cross-language plagiarism detection, and in addressing the existing problems in cross-language plagiarism detection, this paper proposes a cross-language plagiarism detection technique based on fingerprint fusion. The proposed technique is experimentally validated on a manually constructed plagiarism corpus, with detailed and comparative analyses of the experimental results performed to verify its effectiveness.

Full Text

A Cross-Language Plagiarism Detection Technology Based on Fingerprint Fusion

Liu Gang, Zuo Quan, Yang Qianru
(College of Computer Science & Technology, Harbin Engineering University, Harbin 150001, China)

Abstract: Cross-language plagiarism has always been a hotspot for academic misconduct and an extremely difficult behavior to detect. Cross-language plagiarism detection and identification technology is urgently needed and represents the greatest technical challenge in the anti-plagiarism field. Based on a comprehensive review and analysis of current research on monolingual and cross-language plagiarism detection, this paper proposes a cross-language plagiarism detection technology based on fingerprint fusion to address existing problems in cross-language plagiarism detection. The proposed technique is experimentally

validated on artificially constructed plagiarism datasets, with detailed analysis and comparative evaluation of results demonstrating its effectiveness.

Keywords: intermediate fingerprint; fingerprint fusion; semantic disambiguation; cross-language plagiarism detection

0 Introduction

With the rapid development of Internet technology, people can conveniently access various types of information through multiple channels. However, while the network facilitates daily life, it also enables anyone to easily copy others' content as their own, treating existing ideas as innovative contributions, thereby inadvertently forming plagiarism. Plagiarism detection not only protects intellectual property rights but also respects knowledge workers. It has become a research field, particularly in academia, aimed at reducing infringement and studying plagiarism behavior patterns. Researching cross-language plagiarism detection technology for practical system applications is significant for curbing academic misconduct and protecting original authors' rights, thereby promoting academic progress. Alzahrani et al. categorized plagiarism into two types: literal plagiarism, where plagiarists simply copy and paste text without spending much time hiding their academic crimes; and intelligent plagiarism, where plagiarists attempt to deceive readers by transforming others' contributions into their own through translation, synonym replacement, and various intelligent methods to hide, confuse, and alter the original work [1-3]. Documents formed through cross-language plagiarism are essentially incomparable in linguistic form, making automatic detection through software extremely difficult. Currently, research in this area is limited, and no publicly available practical software has emerged. A study by McCabe indicated that among 18,000 students, 40% admitted to plagiarizing at least once, including cross-language plagiarism [4]. Such incidents occur not only abroad but also domestically, ranging from middle school students to doctoral candidates, and from simple copying to synonym replacement and paraphrasing.

1 Research Status

Research on cross-language plagiarism detection abroad has only recently emerged and is in a rapid development phase. In 2008, Alberto et al. [5] used statistical models for cross-language plagiarism detection, relying on statistical bilingual dictionaries generated from parallel corpora and bilingual alignment algorithms. Also in 2008, Ceska et al. [6] proposed MLPlag, a cross-language plagiarism detection method based on word positions. This method employed EuroWordNet to convert words into a language-independent representation, and the authors built two multilingual corpora: JRC-EU and fairy tales. In 2007, Potthast et al. [7] introduced a new multilingual retrieval model—Cross-Language Explicit Semantic Analysis—to analyze cross-language similarity, conducting experiments on both multilingual parallel corpora

(JRC-Acquis) and multilingual comparable corpora (Wikipedia). In 2010, they compared this method with others and found that character n-grams achieved better results, though character N-Gram-based methods are unsuitable for comparisons between grammatically unrelated language pairs [8]. In 2009, Pinto et al. [9] addressed the issue where machine translation and plagiarism detection were always treated as separate steps in cross-language plagiarism detection based on machine translation, along with accumulated inaccuracies, by proposing a method that directly integrates these two steps to improve detection accuracy. In 2010, Pereira et al. [10] proposed a new method for cross-language plagiarism detection consisting of five main stages: language normalization, candidate document retrieval, classifier training, plagiarism behavior analysis, and post-processing. In 2013, Alberto et al. [11] proposed a free and usable cross-language plagiarism detection framework, exploring the applicability of three cross-language similarity evaluation models: Cross-Language Alignment-based Similarity Analysis (CL-ASA), Cross-Language N-Gram-based plagiarism detection (CL-CNG), and Machine Translation plus Monolingual Analysis (T+MA). In the same year, Marc Franco-Salvador et al. [12] proposed a cross-language plagiarism detection method based on multilingual semantic networks. This method used BabelNet, a directed graph whose nodes represent multilingual concepts and named entities and edges represent semantic relationships between them. To verify its effectiveness, they conducted experiments on German-English and Spanish-English plagiarism corpora from PAN 2011, with results showing that the multilingual semantic network-based approach, being language-independent, outperformed other methods. In 2014, Aljohani et al. [13] applied the Winnowing algorithm to detect cross-language plagiarism between Arabic and English, conducting experiments on Wikipedia using precision and recall as evaluation metrics.

Cross-language plagiarism detection research abroad is in its early stages of rapid development, while domestic research in this area remains scarce. Existing foreign studies cover Arabic-English, German-English, etc., but due to Chinese particularities, some foreign cross-language plagiarism detection techniques are not applicable domestically. Since words often have multiple meanings and linguistic differences are especially pronounced compared to Chinese, plagiarism formed by directly translating from English literature into Chinese is difficult to detect. Addressing this problem, this paper researches cross-language plagiarism detection technology to achieve cross-language text plagiarism detection.

2.1 Digital Fingerprint Concepts

A digital fingerprint is a digital code generated by hashing certain features of text through specific selection strategies [14]. Direct string matching on original text presents many problems, such as large storage space, low efficiency, and insufficient accuracy. Therefore, text must be mapped into fingerprints for plagiarism detection.

To evaluate the degree of text plagiarism, similarity between two text finger-

prints must be calculated, as fingerprints should adequately represent the text. Based on digital fingerprint definitions, several factors must be considered when generating fingerprints: text block granularity, fingerprint selection strategy, text quantity, and hash function selection [15].

Text granularity refers to the text length used for generating digital fingerprints. The choice of text granularity significantly impacts plagiarism detection accuracy. The maximum granularity is the entire text, which can only detect verbatim copy-pasting but cannot identify slightly modified text. The minimum granularity is a single character, which leads to excessive fingerprint generation, slow efficiency, and many false matches, reducing precision. Fingerprint selection strategies include full fingerprint selection, frequency-based selection, structure-based selection, and position-based selection. The text block selection issue is directly related to fingerprint quantity: too many fingerprints yield high accuracy but require large computational and storage overhead. Therefore, an appropriate fingerprint quantity must be selected for computation.

2.2 WordNet-Based Intermediate Fingerprint Encoding

WordNet is a machine-readable dictionary based on cognitive linguistics established by Princeton University. WordNet describes objects including English compounds, phrasal verbs, collocations, idioms, and words. The basic unit is the word; WordNet contains no structural units smaller than words nor organizational units larger than words. It differs from both traditional dictionaries and thesauruses, being a hybrid that combines features of both types. The fingerprint generation process maps selected strings to unique integers. Since noun correspondences are most explicit across languages and most article content consists of nouns, and nouns have the clearest tree structure in WordNet, this algorithm only performs fingerprint encoding on nouns.

Nouns in WordNet are organized in a tree structure, with all nouns under a tree rooted at (entity.n.01). Semantic relations are WordNet' s most important relationships, and word senses are fundamental components, represented in WordNet as synsets. The tree structure in WordNet uses synsets as nodes, with each node representing a semantic meaning or concept. The hypernym/hyponym relationship in WordNet is the parent-child node relationship, where parent nodes (hypernyms) generally represent more abstract meanings than child nodes. From root to leaf nodes, word senses become increasingly specific and specialized. [Figure 1: see original paper] shows part of the noun synset tree structure in WordNet 2.1.

Since many languages have WordNet versions corresponding to English WordNet, this overcomes language barriers to some extent, as each synset becomes a language-independent semantic node when different language WordNets are aligned. These semantics can only be appropriately expressed through some natural language. In WordNet 3.0, there are 117,659 synsets total, with 82,115 noun synsets, accounting for approximately 80% of all synsets. This paper encodes all

noun synsets, generating fingerprints that form a language-independent semantic intermediate layer, hence called intermediate fingerprints. Considering the need for subsequent noun semantic disambiguation and fingerprint extraction, and to improve efficiency, Algorithm 1 uses the following approach to encode noun synsets in WordNet:

a) Child node codes use parent node codes as prefixes;

b) Each level is encoded with a certain number of binary bits, where the number is the maximum child count at that level;

c) Encoding proceeds from the highest level, using 1 to n-bit binary codes for the first level, n+1 to 2n-bit binary for the second level, and so on.

[Figure 2: see original paper] illustrates the intermediate fingerprint encoding algorithm.

The intermediate fingerprint encoding algorithm is as follows:

Algorithm 1: Intermediate Fingerprint Encoding
Input: Preprocessed text
Output: Intermediate fingerprints of WordNet noun synsets after encoding

```
Define an empty queue
Add the root node of the WordNet noun tree structure to the queue
Define a dictionary
while queue length > 0:
    node = queue.pop(0)
    for child in node.hyponyms():
        Add child to queue
        dict[node][level][codeLen] = [1, 2, 4, 6, 9, 7, 8, 9, 9, 9, 9, 7, 7, 5, 5, 6, 5, 4,
        dict[node][level][codeLen][count] = (len(child.hyponyms()), code_str)
        count += 1
for x in dict:
    dict[x][2] = dict[x][2] + {129, 10111819, ...}
```

The left-aligned binary representation is extended to 117 bits, with insufficient bits padded with zeros.

2.3 Text Preprocessing

Since hashing is performed on nouns, Chinese and English texts must be preprocessed to extract the required nouns.

ICTCLAS is used for Chinese text segmentation and part-of-speech tagging, after which nouns are extracted and stored in a list. The Chinese lexical analysis system ICTCLAS, developed by the Chinese Academy of Sciences, achieves over 95% segmentation accuracy based on years of research and won first place in evaluations organized by the 973 Expert Group. It is based on a cascaded hidden

Markov model (CHMM). This paper uses the Stanford log-linear part-of-speech tagger developed by the Stanford NLP Group for English text POS tagging and lemmatization. This tool tags English text through a bidirectional dependency network, effectively utilizing prior conditions in linear models and considering lexical relationships between consecutive words, achieving 97.24% POS tagging accuracy.

2.4 Semantic Disambiguation Based on Intermediate Fingerprints

After text preprocessing, a noun sequence is obtained. Due to polysemy, the meaning of each word in context must be determined. Although some text preprocessing considers semantic disambiguation, it only works in monolingual contexts and cannot handle cross-language scenarios. This paper employs a semantics-independent disambiguation algorithm based on intermediate fingerprints for noun sequence disambiguation—primarily using concept relevance principles to select multiple word senses within a disambiguation window and calculating semantic density for disambiguation. The disambiguation result is the synset under the subtree with maximum semantic density.

Assuming a disambiguation window size of 19, the window contains extracted nouns, with the middle word being the target for disambiguation. For example: [r , r , ···, r , r , r , ···, r], where r is the word to be disambiguated. After determining each word' s sense, the window moves back by one position, making r the target word, and so on until all nouns are processed. For a window of length n, with the target word at position i, the main steps of the disambiguation algorithm are:

a) Merge synsets containing each r in the window into a large candidate set C;

b) Sort all synsets in C by their intermediate fingerprints;

c) Calculate semantic density for any combination of synsets in C, with the disambiguation result being the synset under the subtree with maximum semantic density;

d) Move the window back by one position and repeat until all nouns are disambiguated.

The semantic disambiguation algorithm based on intermediate fingerprints is as follows:

Algorithm 2: Semantic Disambiguation Based on Intermediate Fingerprints
Input: Noun sequence extracted from text paragraph
Output: Fingerprint sequence of determined senses after disambiguation

```
while sense count < noun sequence length:
    MaxD = 0
    R = []
    for i in range(0, p):
        for j in range(i+1, p+1):
            C' = {r_i, ..., r_j}
            if |C'| == 1 or |C'| == 2:
                D = semantic_density(C')
                if D > MaxD:
                    MaxD = D
                    sense = C'
    sense count += 1
    return fingerprint sequence
```

2.5 Fingerprint Selection

Fingerprint selection directly impacts subsequent similarity detection and plagiarism detection accuracy. Correct fingerprint selection can fully represent the document, whereas incorrect selection may cause significant deviation from the original. Full fingerprint selection is the simplest approach, offering higher detection accuracy and performance than other strategies, but efficiency drops noticeably for large files. Position-based selection depends on fingerprint location, yielding low accuracy for plagiarism with altered order. Structure-based selection considers paper structure, performing poorly on intelligent plagiarism and cross-language cases with different linguistic structures. Frequency-based fingerprint selection filters overly broad, high-frequency fingerprints and selects appropriately frequent ones.

Since this paper extracts nouns as features, but some nouns are too common and unrepresentative, they must be filtered to select appropriate noun fingerprints as document fingerprints. This paper employs a flexible approach to solve this problem. In WordNet' s noun synset tree structure, upper-level nodes have broader semantics while lower-level nodes are more specific. The depth of synsets in the tree structure serves as a condition for filtering feature sets because, across languages, the overall frequency of noun semantics usage is roughly similar, differing only in specific forms. Additionally, based on the relationship between average semantics and depth, global frequency increases with depth for the first four levels and decreases from the fifth to twentieth levels. Since nodes at shallow depths have overly broad semantics with weak discriminative power, these feature values are filtered out—specifically, fingerprints whose lower 100 bits are all zeros are removed, leaving the document's corresponding fingerprints.

The specific fingerprint selection algorithm is as follows:

Algorithm 3: Fingerprint Selection
Input: Chinese text D , English text D
Output: Chinese fingerprint S , English fingerprint S

```
for each paragraph d  in D :
    for each paragraph d  in D :
        if intersection(d , d ) == 1 or 2:
            Segment and POS-tag d , store in sequence list_ch
            for each item in list_ch:
                if POS is noun:
                    Add to sequence non_ch
            Call Algorithm 2 to disambiguate non_ch
            if lower 100 bits of fingerprint are not all zeros:
                Add to S
            Perform same operations for d  to form S
return S , S
```

Cross-language plagiarism typically uses open-source translation software to translate text before pasting it into one' s own paper. When detecting cross-language plagiarism, it is impossible to analyze all source texts in detail, so source texts must first be retrieved to identify potentially plagiarized paragraphs for detailed analysis.

Thus far, the first part of cross-language plagiarism detection—extracting potential plagiarized document sets—has been introduced, with the core being cross-language text similarity calculation. [Figure 3: see original paper] illustrates the complete process of cross-language fingerprint similarity calculation, using Chinese-English as an example.

3 Detailed Plagiarism Detection Results

3.1 SimWin Fingerprint Fusion Algorithm

Cross-language plagiarism generally involves translation with minor modifications. More refined analysis and calculation are needed to definitively identify plagiarism. Due to structural and expressive differences between languages, this paper cannot achieve word-level precision or character-level detection like monolingual plagiarism detection. Since plagiarists typically translate at the sentence level, sentences serve as the basic unit for cross-language plagiarism detection.

According to the above description, the SimWin algorithm proceeds as follows: first, all texts are segmented into sentences; then, Hamming distance is calculated between sentences, and similarity is computed using the Winnowing algorithm; finally, the final similarity between two sentences is calculated using the fingerprint fusion formula. The specific fingerprint fusion algorithm is shown below:

Algorithm 4: SimWin Fingerprint Fusion
Input: Text paragraph D , text paragraph D
Output: Similarity between sentence pairs after fingerprint fusion

```
Segment D  into sentence sequence S
Segment D  into sentence sequence S
```

```
for each sentence s  in S :
    Generate SimHash fingerprint simhash_f
    Generate Winnowing fingerprint winnowing_f
    for each sentence s  in S :
        Generate SimHash fingerprint simhash_f
        Generate Winnowing fingerprint winnowing_f
        Calculate Hamming distance H(s , s )
        Calculate similarity S(s , s ) using Winnowing
        Calculate fused similarity f(s , s ) using formula
return fused similarities
```

The fusion formula is:

$$f_\alpha(A,B) = (1 - H(A,B)) \times \alpha + S_{\text{winnowing}}(A,B) \times (1-\alpha)$$

where $H(A,B)$ is the Hamming distance between sentences A and B, $S_{\text{winnowing}}(A,B)$ is the similarity calculated by the Winnowing algorithm, α is the weight for SimHash results, and $1-\alpha$ is the weight for Winnowing results. Specific values are determined experimentally for optimal similarity measurement.

SimHash [17] is a locality-sensitive hashing algorithm. Traditional fingerprint algorithms cannot measure text similarity through fingerprints, whereas SimHash can measure similarity via Hamming distance between fingerprints. SimHash is commonly used for massive text similarity calculation, as its core idea reduces dimensionality by hashing high-dimensional feature vectors into fixed-length fingerprints for similarity comparison, though this efficiency comes at the cost of reduced precision.

The Winnowing algorithm has excellent noise reduction, extracting text feature sequences through sliding windows. Its advantages include stability when texts undergo minor changes (i.e., when hash sequences change slightly, extracted feature sequences remain largely unchanged), and window size has minimal impact on detection results for minor text variations, enhancing algorithm robustness. However, it suffers from excessive feature fingerprints and incomplete fingerprint selection.

This paper comprehensively considers the strengths and weaknesses of SimHash and Winnowing algorithms, noting their different fingerprint extraction emphases, and fuses both algorithms to propose the SimWin algorithm for more accurate sentence similarity calculation. Compared to using a single fingerprint algorithm, the proposed sentence similarity calculation method based on fingerprint fusion integrates both algorithms' characteristics, yielding more accurate and robust results. [Figure 4: see original paper] shows the detailed plagiarism detection analysis process.

3.2 Plagiarism Fragment Merging

Through sentence similarity calculation and threshold filtering, we can determine whether sentences in suspicious texts plagiarize sentences from source documents. Sentences are used as the basic unit because structural differences between languages prevent character-level detection like monolingual cases. However, two consecutively plagiarized sentences may appear, which should be reported as a single detection result rather than two, as shown in [Figure 5: see original paper].

[Figure 5: see original paper] shows analysis results for Document A. The detection method identifies two plagiarized sentences in the suspicious text, both plagiarized from Document B. The first plagiarism starts at character position 1000 in the suspicious text, copying 500 characters, while the corresponding source text starts at position 3000 in Document B. The second detection shows another plagiarism immediately following the first. These should be connected rather than reported separately. The merged result is shown in [Figure 6: see original paper].

To merge consecutive plagiarized sentences, this paper uses the following method:

a) Perform collective detection by classifying source texts by the attribute source_reference;

b) For each set obtained in (a), sort them by attribute this_offset in ascending order;

c) Connect adjacent detections separated by at most a predefined number of characters;

d) Report only one plagiarism detection result for each plagiarized paragraph (selecting the source document with the longest paragraph), ensuring no more than one possible plagiarism source per paragraph in the suspicious text.

Following these steps merges plagiarism results into integrated rather than fragmented detections.

4 Experimental Results Analysis

Cross-language plagiarism detection experiments primarily use simplified Chinese and English as test subjects. Due to the lack of suitable Chinese-English cross-language plagiarism corpora, this paper uses Chinese and English abstracts of master' s and doctoral theses downloaded from CNKI as the base corpus, plus 10 English papers translated into Chinese via Google Translate, subsequently modified through manual addition and deletion to form the experimental plagiarism corpus. The dataset includes 5,000 Chinese texts and 5,000 English texts,

all stored in (*.txt) format. Experiments on these 10,000 texts yield relevant data for analysis. The experiments consist of three phases:

a) Detailed analysis of WordNet' s noun tree structure, encoding synset nodes into fingerprints using the intermediate fingerprint encoding algorithm, and preprocessing Chinese and English texts to extract noun sequences using natural language processing techniques.

b) Semantic disambiguation of noun sequences based on intermediate fingerprints, extracting paragraph-level Chinese and English fingerprints through fingerprint selection strategies, calculating similarity using the Dice coefficient, and selecting potential plagiarized paragraphs through threshold filtering.

c) Translating Chinese texts into corresponding English texts via Google API, segmenting texts into sentences, calculating sentence similarity using the SimWin algorithm, selecting plagiarized sentences through thresholds, and finally merging plagiarism fragments to form final detection results.

4.1 Intermediate Fingerprint Encoding

This experiment uses Princeton WordNet 3.0 for English and the corresponding Chinese Open Wordnet [18] established by Nanyang Technological University, which are aligned. After encoding WordNet noun synsets, they are stored as quadruples: {synset; English words; Chinese words; intermediate fingerprint}. [Figure 7: see original paper] shows partial intermediate fingerprint encoding results.

In [Figure 7: see original paper], Synset('message.n.01') represents a synset, where message.n.02 is the second sense of "message" as a noun. The following line shows English words in this sense, the next line shows Chinese words (or None if unavailable), and the final line shows the 117-bit binary intermediate fingerprint after encoding. The results show each synset represents a sense that may contain multiple English and Chinese words (85% have corresponding words, minimally impacting subsequent experiments).

4.2 Potential Plagiarism Document Retrieval

After semantic disambiguation and fingerprint selection, Chinese and English text fingerprints are formed, and similarity is calculated using the Dice coefficient. shows partial Chinese-English text similarity calculation results.

TABLE:1 Partial Chinese-English Text Similarity Calculation Results

Chinese Text	English Text	Similarity
zh00001	en00001	0.73
zh00001	en00002	0.21

Chinese Text	English Text	Similarity
zh00001	en00003	0.18
...	...	...

The table shows "zh00001.txt" and "en00001.txt" have the highest similarity. Manual analysis confirms they are mutual translations, validating the reasonableness of Chinese-English text similarity calculation. To better evaluate the algorithm, precision (P), recall (R), and F-measure are used:

$$P = \frac{TP}{TP + FP} \times 100\%$$

$$R = \frac{TP}{TP + FN} \times 100\%$$

$$F\text{-measure} = \frac{2PR}{P + R}$$

where TP is the number of texts correctly detected as similar, FP is the number of texts incorrectly detected as similar, and FN is the number of similar texts not detected. [Figure 8: see original paper] shows the average precision, recall, and F-measure at different similarity thresholds.

The results show the optimal F-measure (0.7214) occurs at similarity threshold 0.71, balancing precision and recall. Since this process only extracts nouns as feature sequences and requires context consideration, it works for longer texts but precision drops for shorter paragraphs and sentences, necessitating subsequent detailed analysis.

4.3 SimWin Algorithm Experimental Analysis

In detailed analysis, suspicious Chinese texts are translated into English via Google API, then monolingual similarity analysis is performed. For detailed comparison, the fingerprint fusion algorithm operates at the sentence level, using periods as sentence delimiters. Fifty sentences with average length of 30 words are selected from the plagiarism set as the test set. Precision and recall are calculated for each sentence at different thresholds, then averaged. [Figure 9: see original paper] shows the relationship between SimHash threshold and precision/recall.

[Figure 9: see original paper] shows precision and recall reach a balance at threshold 10, which is therefore selected as the SimHash threshold.

The two algorithms are fused to measure sentence similarity, but the formula' s α parameter requires experimental determination as optimal values vary across environments. Using the same test set as SimHash and Winnowing, the F-measure determines the optimal α and similarity threshold. [Figure 10: see

original paper] shows the relationship between α, similarity threshold, and F-measure.

The results show when $\alpha = 0.3$, F-measure is generally higher than other values, with the highest F-measure at similarity threshold 0.4.

TABLE:2 shows SimHash Hamming distances, Winnowing similarities, and fused similarities for one sentence against ten other sentences.

TABLE:2 Partial Sentence Similarity Calculation Results

Sentence	SimHash Distance	Winnowing Similarity	Fused Similarity
sen00005	8	0.42	0.51
sen00011	15	0.18	0.23
sen00020	9	0.38	0.45
...	...	...	...

With Hamming distance threshold 10, SimHash detects two potential plagiarized sentences, while Winnowing with threshold 0.35 detects four. The suspicious sentence is actually a translation of English sentences "sen00005" , "sen00020" , "sen00086" , and "sen00310" . Since cross-language plagiarism typically involves translation, these four sentence pairs are indeed plagiarism cases. Winnowing' s precision is higher than SimHash' s, but some similar sentences have low similarity scores, which is unfavorable for plagiarism detection. The fused approach improves detection capability.

[Figure 11: see original paper] compares the three sentence similarity calculation methods. The proposed fingerprint fusion algorithm outperforms SimHash and Winnowing in precision and F-measure, with recall comparable to Winnowing, confirming its effectiveness.

In [11], Alberto et al. compared CL-ASA, CL-CNG, and T+MA methods under the same experimental conditions, finding T+MA (machine translation plus monolingual analysis) most effective. Their experiments used Google API for translation, TF-IDF for term weighting, and cosine similarity on bag-of-words models, conducted between English, German, and Spanish, rarely including Chinese. This paper compares T+MA, CL-ASA, and the proposed fingerprint fusion method. [Figure 12: see original paper] shows the comparison results.

[Figure 12: see original paper] shows the proposed method achieves higher precision (0.87) and recall (0.78) than the other two methods in this experimental environment, validating the effectiveness of the fingerprint fusion-based cross-language plagiarism detection technology.

Since fingerprint-based methods are efficient, this paper also compares detection time across different corpus sizes. [Figure 13: see original paper] shows the time comparison.

[Figure 13: see original paper] shows little time difference between the proposed method and T+MA for small texts, but the proposed method is significantly faster for larger texts. CL-ASA initially requires less time as it builds statistical translation models first. Since the proposed method uses bit operations for similarity calculation, with subsequent disambiguation and fingerprint selection built upon this, efficiency is greatly improved. The fingerprint fusion approach combines two algorithms, enhancing both precision and recall.

5 Conclusion

With Internet development and machine translation advances, plagiarism has become increasingly severe. Mature monolingual plagiarism detection technology has led some to engage in cross-language plagiarism, which is more complex than monolingual detection due to linguistic and structural inconsistencies. This paper proposes a cross-language plagiarism detection method based on fingerprint fusion, completing the following work:

a) Analyzed the objectives and significance of cross-language plagiarism detection, reviewed digital fingerprint technology in monolingual detection and current cross-language methods, and proposed establishing intermediate fingerprints on WordNet using fingerprint technology for cross-language similarity calculation to address existing limitations.

b) Analyzed WordNet structure, particularly the noun synset tree, and presented an intermediate fingerprint encoding algorithm based on this tree structure, where child nodes use parent prefixes as prefixes, representing semantic breadth and specificity in binary form to overcome language barriers.

c) Investigated text preprocessing technologies, selecting appropriate techniques including segmentation and POS tagging. Semantic disambiguation based on intermediate fingerprints determines the unique sense of features in context, followed by fingerprint extraction based on semantic frequency to filter overly broad words, forming text-specific fingerprints.

d) Proposed a fingerprint fusion-based plagiarism detection method. Potential plagiarism document sets are retrieved through cross-language paragraph similarity search using intermediate fingerprints, followed by sentence-level detailed analysis using the SimWin algorithm. Fingerprint fusion combines SimHash and Winnowing algorithms after analyzing their respective strengths, improving final result accuracy and robustness. Plagiarism fragment merging forms final detection results, with experiments validating the method' s effectiveness.

Cross-language plagiarism detection remains a challenge in natural language processing, and this algorithm offers one solution. However, due to objective constraints, several issues warrant further improvement. First, WordNet collections are incomplete across languages, encountering out-of-vocabulary words

that are currently ignored; whether this impacts results requires further study. Second, since nouns have the clearest structure in WordNet, only nouns are extracted, ignoring other POS tags' contributions. Introducing other words would complicate encoding. While suitable and efficient for longer paragraphs, the algorithm' s precision drops for shorter paragraphs and sentences, requiring translation at the sentence level. Whether translating suspicious documents into source language or vice versa impacts results also needs further investigation.

References

[1] Alzahrani S M, Salim N, Abrabam A. Understanding plagiarism linguistic patterns, textual features and detection methods [J]. IEEE Trans on Systems, Man and Cybernetics. 2012, 42 (2): 133-149.

[2] Potthast M, Eiselt A, Barrón-Cedeño A. Overview of the 3rd international competition on plagiarism detection [C]// Notebook Papers of CLEE Labs and Workshop. 2011: 19-22.

[3] Soleman S, Purwarianti A. Experiments on the Indonesian plagiarism detection using latent semantic analysis [C]// Proc of International Conference on Information and Communication Technology. 2014: 5579-5583.

[4] Potthast M, Barrón-Cedeño A, Stein B. Cross-language plagiarism detection [J]. Language Resources & Evaluation, 2011, 45 (1): 45-62.

[5] Barrón-Cedeño A, Rosso P, Pinto D, Juan A. On cross-lingual plagiarism analysis using a statistical model [C]// Proc of ECAI PAN Workshop. 2008: 23-27.

[6] Ceska Z, Toman M, Jezek K. Multilingual plagiarism detection [C]// Proc of the 13th International Conference on Artificial Intelligence: Methodology, Systems, and Applications. 2008: 83-92.

[7] Pouliquen B, Steinberger R, Ignat C. Automatic identification of document translations in large multilingual document collections [C]// Proc of International Conference on Advances in Natural Language Processing. 2003: 401-408.

[8] Potthast M. Wikipedia in the pocket: indexing technology for near-duplicate detection and high similarity search [C]// Proc of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2007: 900-909.

[9] Potthast M, Stein B, Barrón-Cedeño A, et al. An evaluation framework for plagiarism detection [C]// Proc of the 23rd International Conference on Computational Linguistics: Posters. 2010: 997–1005.

[10] Pinto D, Civera J, Barrón-Cedeño A, et al. A statistical approach to crosslingual natural language tasks [J]. Journal of Algorithms, 2009, 64 (1): 67-80.

[11] Pereira R C, Moreira V P, Galante R. A new approach for cross-language plagiarism analysis [C]// Proc of International Conference of the

Cross-Language Evaluation Forum. 2010: 15-26.

[12] Barrón-Cedeño A, Gupta P, Rosso P. Methods for cross-language plagiarism detection [J]. Knowledge-Based Systems, 2013, 50 (9): 211-217.

[13] Marc F, Parth G, Paolo R. Cross-language plagiarism detection using a multilingual semantic network [C]// Proc of European Conference on Information Retrieval. Berlin: Springer, 2013: 710-714.

[14] Adel A, Masnizah M. Arabic-English cross-language plagiarism detection using winnowing algorithm [J]. Information Technology Journal. 2014, 13 (14): 2349-2355.

[15] Gharghe Z E, Bidgoli B M. Weighted shingling: an adaptation of shingling for weighted shingles [C]// Proc of International Conference on Innovations in Information Technology. 2009: 150-154.

[16] (Reference omitted in original)

[17] (Reference for SimHash omitted in original)

[18] Chinese Open Wordnet, Nanyang Technological University.

[19] (Reference omitted in original)

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