Understanding Plagiarism Linguistic Patterns, Textual Features, and Detection Methods

Salha M. Alzahrani, Naomie Salim, and Ajith Abraham, *Senior Member, IEEE*

***Abstract*—Plagiarism can be of many different natures, ranging from copying texts to adopting ideas, without giving credit to its originator. This paper presents a new taxonomy of plagiarism that highlights differences between literal plagiarism and intelligent plagiarism, from the plagiarist's behavioral point of view. The taxonomy supports deep understanding of different linguistic patterns in committing plagiarism, for example, changing texts into semantically equivalent but with different words and organization, shortening texts with concept generalization and specification, and adopting ideas and important contributions of others. Different textual features that characterize different plagiarism types are discussed. Systematic frameworks and methods of monolingual, extrinsic, intrinsic, and cross-lingual plagiarism detection are surveyed and correlated with plagiarism types, which are listed in the taxonomy. We conduct extensive study of state-of-the-art techniques for plagiarism detection, including character *n*-gram-based (CNG), vector-based (VEC), syntax-based (SYN), semantic-based (SEM), fuzzy-based (FUZZY), structural-based (STRUC), stylometric-based (STYLE), and cross-lingual techniques (CROSS). Our study corroborates that existing systems for plagiarism detection focus on copying text but fail to detect intelligent plagiarism when ideas are presented in different words.**

***Index Terms*—Linguistic patterns, plagiarism, plagiarism detection, taxonomy, textual features.**

I. Introduction

The problem of plagiarism has recently increased because of the digital era of resources available on the World Wide Web. Plagiarism detection in natural languages by statistical or computerized methods has started since the 1990s, which is pioneered by the studies of copy detection mechanisms in digital documents [42], [43]. Earlier than plagiarism detection in natural languages, code clones and software misuse detection has started since the 1970s by the studies to detect programming code plagiarism in Pascal and C [28], [44]–[47]. Algorithms of plagiarism detection in natural languages and programming languages have noticeable differences. The first one tackles different textual features and diverse methods of detection, while the latter mainly focuses on keeping track of metrics, such as number of lines, variables, statements, subprograms, calls to subprograms, and other parameters. During the last decade, research on automated plagiarism detection in natural languages has actively evolved, which takes the advantage of recent developments in related fields like information retrieval (IR), cross-language information retrieval (CLIR), natural language processing, computational linguistics, artificial intelligence, and soft computing. In this paper, a survey of recent advances in the area of automated plagiarism detection in text documents is presented, which started roughly in 2005, unless it is noteworthy to state a research prior than that. Earlier study was excellently reviewed by [48] and [52]–[55].

This paper brings patterns of plagiarism together with textual features for characterization of each pattern and computerized methods for detection. The contributions of this paper can be summarized as follows: First, different kinds of plagiarism are organized into a taxonomy that is derived from a qualitative study and recent literatures about the plagiarism concept. The taxonomy is supported by various *plagiarism patterns* (i.e., examples) from available corpora for plagiarism [60]. Second, different *textual features* are illustrated to represent text documents for the purpose of plagiarism detection. Third, *methods* of candidate retrieval and plagiarism detection are surveyed, and correlated with plagiarism types, which are listed in the taxonomy.

The rest of the paper is organized as follows: Section II presents the taxonomy of plagiarism and linguistic patterns. Section III explores the concept of plagiarism detection system in various ways: black-box versus white-box designs, extrinsic versus intrinsic tasks, and monolingual versus cross-lingual systems. Section IV demonstrates various textual features to quantify documents as a *proviso* in plagiarism detection. Section V discusses the analogous research between extrinsic plagiarism detection and IR, and between cross-language plagiarism detection and CLIR, and illustrates plagiarism detection methods, including character n-gram-based (CNG), vector-based (VEC), syntax-based (SYN), semantic-based (SEM), fuzzy-based (FUZZY), structural-based (STRUC), stylometric-based (STYLE), and cross-lingual techniques (CROSS). Section VI maps between methods and types of plagiarism in our taxonomy, and Section VII draws a conclusion for this paper.

II. Plagiarism Taxonomy and Patterns

There are no two humans, no matter what languages they use and how similar thoughts they have, write exactly the same text. Thus, written text, which is stemmed from different authors,

Manuscript received October 23, 2010; revised January 15, 2011; accepted March 19, 2011. Date of publication May 12, 2011; date of current version February 17, 2012. This paper was recommended by Associate Editor Z. Wang.

S. M Alzahrani is with the Faculty of Computer Science and Information Systems, Taif University, Alhawiah, 888 Taif, Saudi Arabia (e-mail: s.zahrani@tu.edu.sa).

N. Salim is with the Faculty of Computer Science and Information Systems, University of Technology Malaysia, Skudai 81310, Malaysia (e-mail: naomie@utm.my).

A. Abraham is with the VSB-Technical University of Ostrava, 70103 Ostrava, Czech Republic, and also with the Machine Intelligence Research Labs, Scientific Network for Innovation and Research Excellence, Seattle, WA 98102 USA (e-mail: ajith.abraham@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCC.2011.2134847

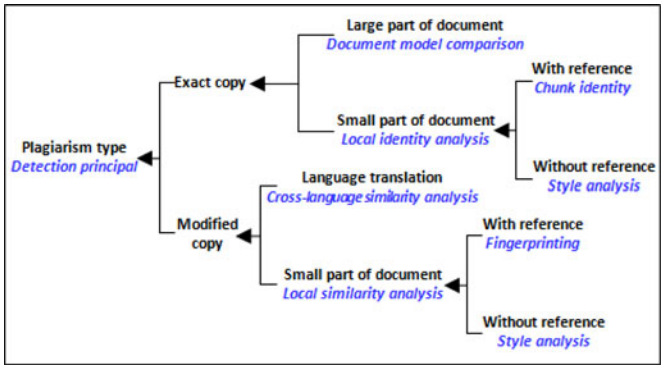

Fig. 1. Plagiarism types with some related detection principles [20].

should be different, to some extent, except for cited portions. If proper referencing is abandoned, problems of plagiarism and intellectual property arise. The existence of academic dishonesty problems has led most, if not all, academic institutions and publishers to set regulations against the offence. Borrowed content of any form require directly or indirectly quoting, in-text referencing, and citing the original author in the list of references [62].

A number of research works have addressed plagiarism in academia [62]–[65] and illustrated different types of plagiarism and available software for plagiarism detection. For example, a recent book [21], [63] provides an extensive linguistic analysis of plagiarism in academic writing. However, little research has related linguistic patterns of plagiarism with computerized textual features and automated techniques for extracting and detecting such types. Eissen *et al*. [33] discussed some plagiarism types with related detection principles, as shown in Fig. 1. This study extends the taxonomy in [33] and relates different types of plagiarism with recent advances of detection methods. To this aim, we conducted a qualitative study at the University of Technology Malaysia. The objectives of the study were to embrace this study with academician's experience, who faces plagiarism, to pursue in-depth information around the offence, and to get the story of current plagiarism practices committed by students. The data were collected via several interviews with several faculty members with 10–20-year teaching expertise at the university. The questions focused on different plagiarism practices by the students. The main output of the qualitative study is a new taxonomy of plagiarism that comprehensively relates different types, as shown in Fig. 2. The taxonomy divides plagiarism into two typical types: *literal plagiarism* and *intelligent plagiarism*, based on the *plagiarist's behavior* (i.e., student's or researcher's way of committing plagiarism).

A. Literal Plagiarism

Literal plagiarism is a common and major practice wherein plagiarists do not spend much time in hiding the academic crime they committed. For example, they simply copy and paste the text from the Internet. Aside from few alterations in the original text (marked as underlined), Fig. 3 shows a pattern of text taken entirely word-for-word from the source without direct quotation.

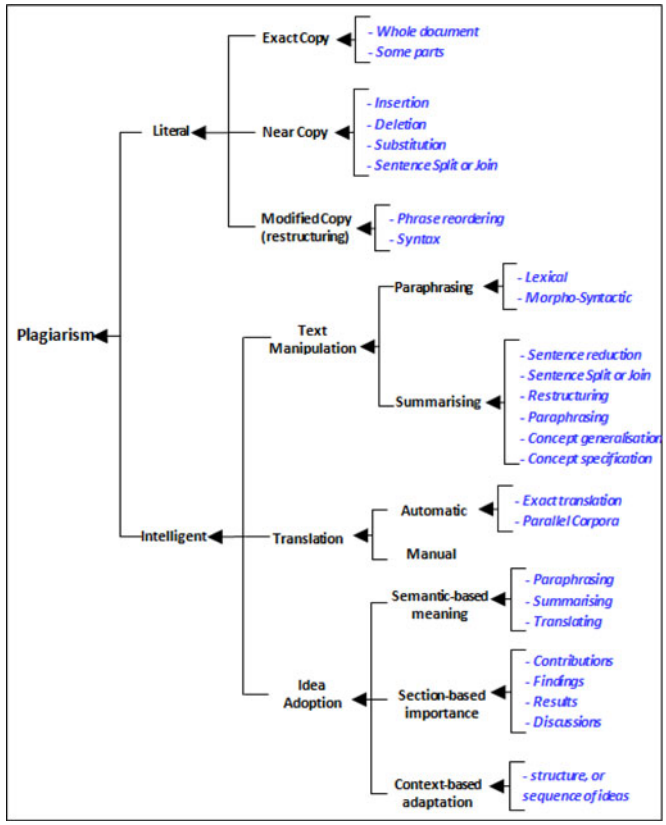

Fig. 2. Taxonomy of plagiarism.

Original: The definition of term includes single words, keywords, or longer phrases [active voice]. If the words are chosen to be the terms, the dimensionality of the vector is the number of words in the vocabulary.

Plagiarized: Keywords, single words or longer phrases are included in the definition of the term [active-passive conversion]. If the words are chosen to be the terms, the dimensionality of the vector is the number of words in the vocabulary.

Fig. 3. Pattern of literal plagiarism extracted from the corpus of plagiarized short answers [61].

In the academic law [62], such practice requires direct quotation around the borrowed content, in-text referencing, and citing the original author in the list of references.

B. Intelligent Plagiarism

Intelligent plagiarism is a serious academic dishonesty wherein plagiarists try to deceive readers by changing the contributions of others to appear as their own. Intelligent plagiarists try to hide, obfuscate, and change the original work in various intelligent ways, including text manipulation, translation, and idea adoption.

1) Text Manipulation: Plagiarism can be obfuscated by manipulating the text and changing most of its appearance. Fig. 4 exemplifies lexical and syntactical paraphrasing, where underlined words are replaced with synonyms/antonyms, and short phrases are inserted to change the appearance, but not the idea, of the text. Paraphrasing while retaining the semantic

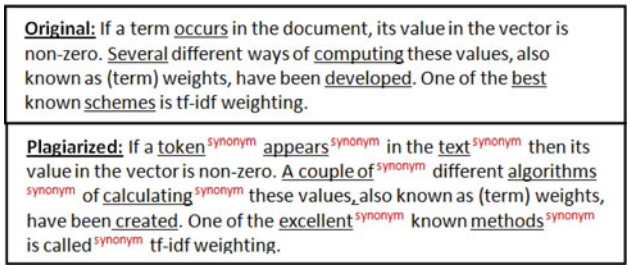

Fig. 4. Pattern of intelligent plagiarism (paraphrasing) extracted from the corpus of plagiarized short answers [61].

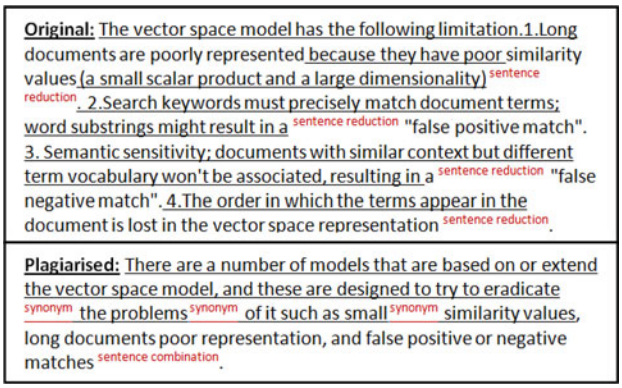

Fig. 5. Pattern of intelligent plagiarism (summarization) extracted from the corpus of plagiarized short answers [61].

meaning requires citations around the borrowed ideas and citing the original author [62], [65].

Besides paraphrasing, summarizing the text in a shorter form using sentence reduction, combination, restructuring, paraphrasing, concept generalization, and concept specification is another form of plagiarism unless it is cited properly. Fig. 5 shows that some sentences are combined and restructured, some phrases are syntactically changed, sentences are reduced by eliminating underlined text in the original text, and synonyms of some words are used in the summary. Although much of the text is changed and fewer phrases are left in the summary, citation and attribution are still required [62].

2) Translation: Obfuscation can also be done by translating the text from one language to another without proper referencing to the original source. Translated plagiarism includes automatic translation (e.g., Google translator) and manual translation (e.g., by people who speak both languages).

Back translated plagiarism [71] is another (easier) form of paraphrasing by automatically translating a text from one language to another and retranslate it back to the first one. Fig 6 shows an example of text translated from English to French and back from French to English. It is obvious that the retranslated text may have poor English, but plagiarists could use spell checkers and other text manipulations to obfuscate plagiarism.

3) Idea Adoption: Idea adoption is the most serious plagiarism that refers to the use of other's ideas, such as results, contributions, findings, and conclusions, without citing the original source of ideas [62]. It is a major offence to steal ideas of others, which is a real academic problem that needs to be investigated.

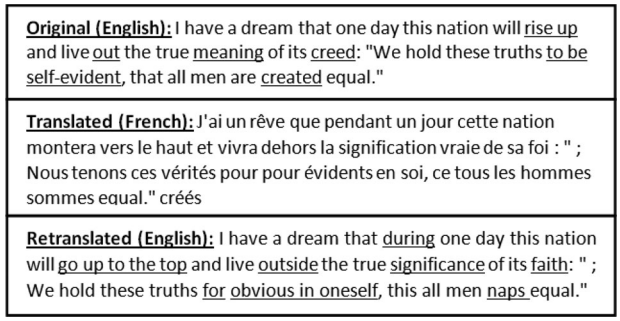

Fig. 6. Back-translated plagiarism [71].

> "What is worse are cases where scientists rewrite previous findings in different words, purposely hiding the sources of their ideas" [73].

Borrowing a few words, but no original ideas, to improve the quality of the English, especially by nonnatives, should not be considered plagiarism [74]. The qualitative study showed that university professors can suspect or detect different types of idea plagiarism using their own expertise. However, computerized solutions for the purpose of detecting idea plagiarism are highly needed, since it is crucial to judge the *quality* of different academic work, including theses, dissertations, journal papers, conference proceedings, essays, and assignments. Idea plagiarism can be classified into three types yet with fuzzy boundaries: semantic-based meaning, section-based importance, and context-based plagiarism.

A narrow view of idea plagiarism can be seen via the *semantic-based meaning* of two texts, e.g., two paragraphs, whereby the same idea is expressed in different words. The semantic-based idea plagiarism can be committed by paraphrasing, summarizing and translating the text.

A deterministic view of idea plagiarism can be seen via the importance of different sections/segments in the documents, i.e., idea plagiarism via *section-based importance* includes plagiarizing substantial segments of a scientific work, such as results, discussions, and findings contributions of the others. Fig. 7 shows an example of literal plagiarism versus idea plagiarism. The example shows that sections, such as the introduction, are of marginal importance and most likely to contain literal but not idea plagiarism, as stated:

> "Copying a few sentences that contain no original idea (e.g., in the introduction) is of marginal importance compared to stealing the ideas of others" [74].

This view of idea plagiarism via section-based importance in scientific articles is supported by many academic institutions:

> "Ethical writing demands that ideas, data, and conclusions that are borrowed from others and used as the foundation of one's own contributions to the literature, must be properly acknowledged" [62].

A holistic view of idea plagiarism can be seen via the *context-based adaptation*, where the author's *structure* of different ideas (e.g., sections, subsections, and logical sequence of ideas), but not necessarily the exact content, is plagiarized from the source. Even if the author rewrites and paraphrases much of the text but maintains the logical sequence of ideas, this practice is

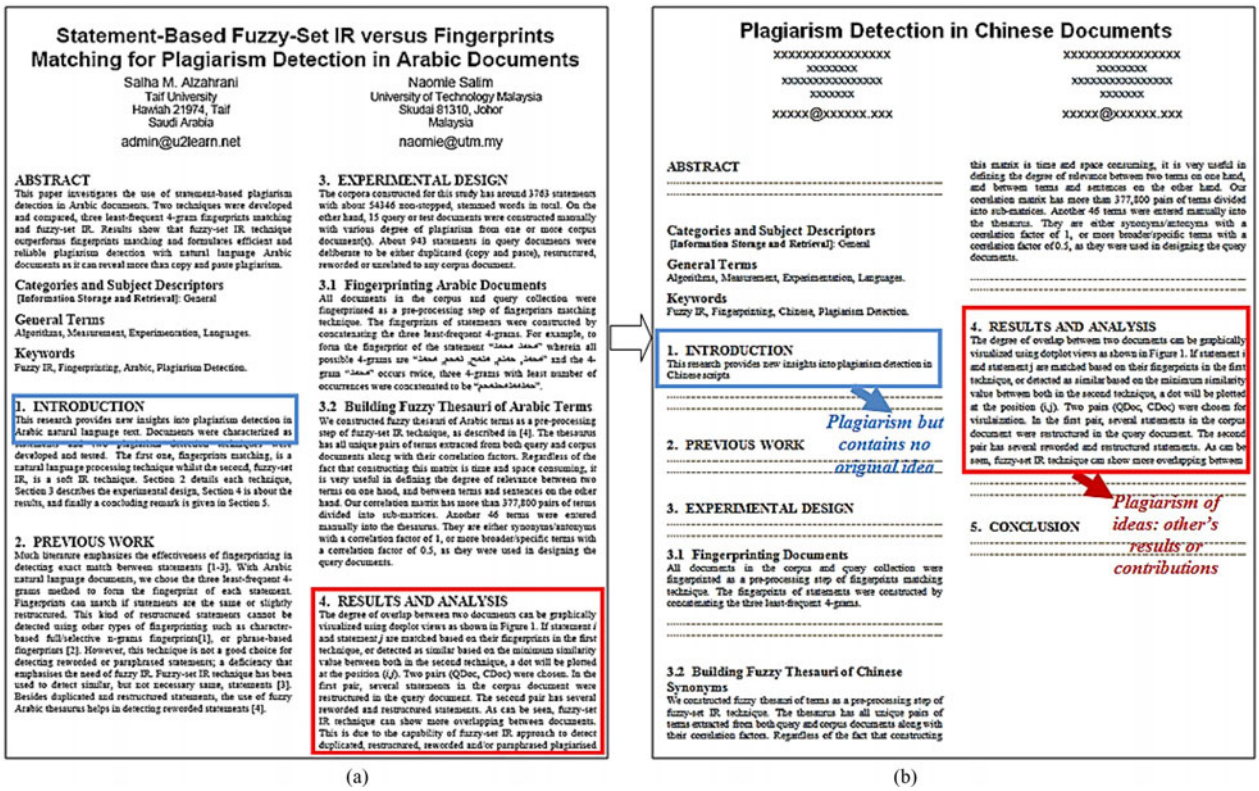

Fig. 7. Literal versus idea plagiarism in a scientific paper. (a) Original paper (Alzahrani and Salim, 2009). (b) Simulated plagiarism based on the importance of different sections/segments in the article.

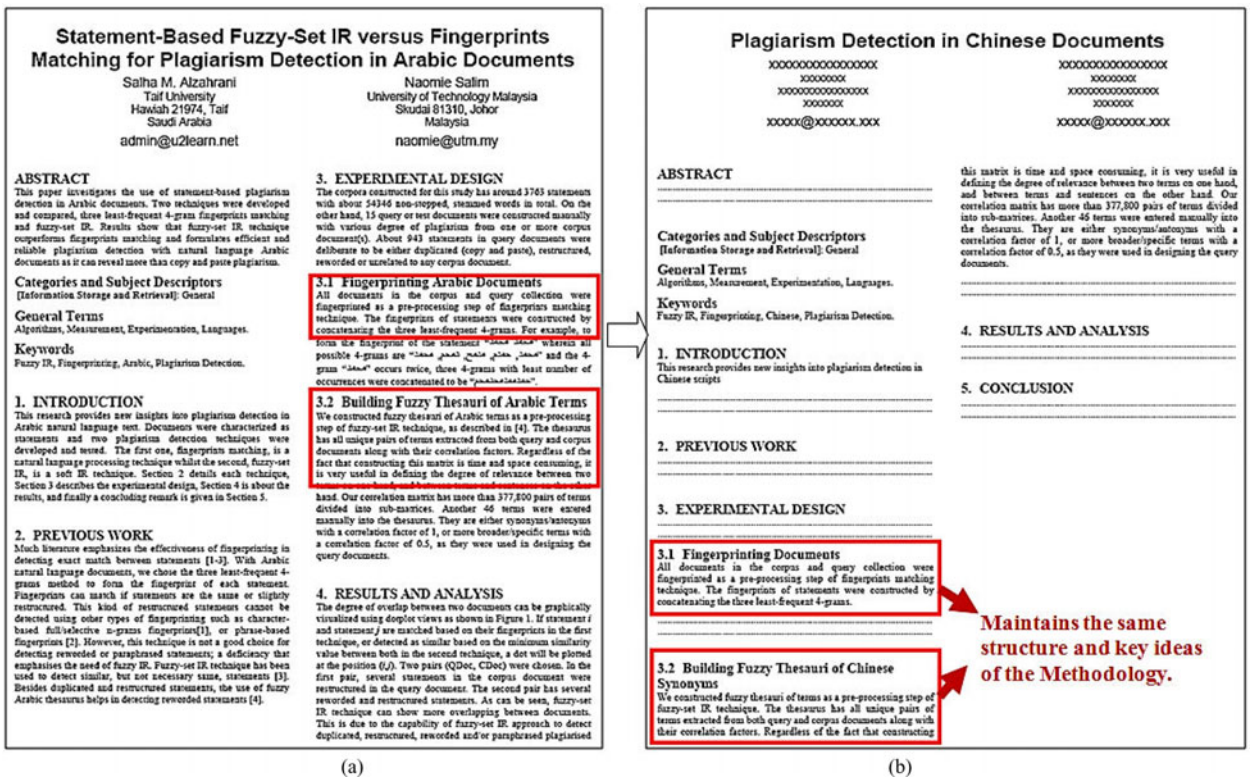

Fig. 8. Pattern of idea plagiarism in a scientific article. (a) Original paper (Alzahrani and Salim, 2009). (b) Simulated plagiarism via the adoption of a sequence of key ideas in the choice of the methodology.

Fig. 9. Black-box design for plagiarism detection system. Items in the brackets are based on the plagiarism detection task.

considered idea plagiarism in some research fields [62]. Fig. 8 shows that the choice of the methodology suggested to solve a particular problem and the basic structure of key ideas are taken, without recognition, from the source. Others' ideas are considered their intellectual property and should not be plagiarized. Rather each author should develop his/her own ideas and give recognition to previous ideas.

III. Plagiarism Detection Systems Framework

Plagiarism detection is the process of featuring the document, analyzing its content, unveiling portions that might be plagiarized, and bringing similar source documents, if they are available. Although humans have the ability to suspect plagiarism by memorizing similar work or by monitoring the writing style, "it requires much effort to be aware of all potential sources on a given topic and to provide strong evidence against an offender" [31]. The need of computerized systems for plagiarism detection is feasible due to human inability to process large documents and to retrieve all suspected parts and original sources.

Among the ways of plagiarism prevention is the use of plagiarism detectors; Turnitin is a very popular one. Academic institutions, publishers, and conference management systems have started to use detectors. Examples include, but not limited to, WCopyFind that is used by the University of Virginia, CrossCheck that is invested by Elsevier, Springer, Massachusetts Institute of Technology press, and other renowned publishers, Docoloc that is integrated with EDAS conference management system. Nowadays, many antiplagiarism tools have been built and can be found on the Internet. Most of them, however, tackle some kinds of textual plagiarism with superficial modifications. Fig. 9 shows the black-box framework for a plagiarism detection system. It has one core input that is expressed as a query/suspicious document d_q, and another optional input, which is the reference collection D, such as the Web. The output is the suspicious fragments/ sections (e.g., paragraphs, statements, etc.), if found with sources of plagiarism, if available. The following sections review many aspects inside the black-box including tasks, languages, methods, and evaluation.

A. Plagiarism Detection Tasks

Plagiarism detection is divided into two formal tasks: extrinsic and intrinsic [25]. Extrinsic plagiarism detection evaluates plagiarism in accordance to one or more source documents. Intrinsic plagiarism detection, on the other hand, evaluates instances of plagiarism by looking into the suspicious/query document in isolation. The first one utilizes the computer's capability in searching large text collection and retrieving possible sources for plagiarism, whereas the second one simulates the human's ability to catch plagiarism via writing style variations.

B. Extrinsic Plagiarism Detection

Extrinsic plagiarism detection is a method of comparing a suspicious document against a set of source collection whereby several text features are used to suspect plagiarism [25]. Much research has been undertaken to tackle this task [1]–[6], [10], [12], [16]–[19], [21], [26], [29], [61], [77]–[102]. Fig. 10(a) shows the white-box design for extrinsic plagiarism detection. The operational framework can be summarized as follows [24], [25]. The inputs are a query document d_q and a reference collection D that may contain the sources of plagiarism. Three main operations are needed (shown in rounded rectangles). First, a small list of candidate documents D_x, which are likely to be sources of plagiarism, are retrieved from D under some retrieval model, as will be discussed in Section V-A. Second, a pairwise feature-based exhaustive analysis is performed to compare d_q with its candidates by using some comparison unit, such as k-grams or sentences, as will be explored in Section V-B. Third, a knowledge-based postprocessing step is performed to merge small detected units into passages or sections and to present the result to a human, who may decide whether or not a plagiarism offense is given. Thus, the final output is pairs of fragments/sections (s_q, s_x), where $s_q \in d_q$, $s_x \in d_x$, and $d_x \in D_x$, such that s_q is pattern of plagiarism from s_x. Note that s_q is one of the plagiarism types that is mentioned in the taxonomy, except translated plagiarism.

C. Intrinsic Plagiarism Detection

Intrinsic plagiarism detection, authorship verification, and authorship attribution are three similar tasks yet with different end goals. In all of them, writing style is quantified and/or feature complexity is analyzed. The different end goals of these tasks are 1) to suspect plagiarism in the intrinsic plagiarism detection; 2) to verify whether the text stems from a specific author or not in the authorship verification; and 3) to attribute the text to authors in the authorship attribution.

> "Intrinsic plagiarism aims at identifying potential plagiarism by analyzing a document with respect to undeclared changes in writing style. Authorship verification aims at determining whether or not a text with doubtful authorship is from an author A, given some writing examples of A, while authorship attribution aims at attributing a document d of unknown authorship, given a set D of candidate authors with writing examples" [23].

That is, intrinsic plagiarism detection can be viewed as the generalization of authorship verification and attribution because intrinsic plagiarism detection analyses the query document in isolation, while authorship analysis problems analyze a document with respect to a set of writing examples of a specific author in authorship verification or a set of candidate authors writing examples in authorship attribution. Many research works have been conducted to tackle the task of intrinsic plagiarism detection [23], [33], [34], [66], [103]–[107]. Fig. 10(b) shows

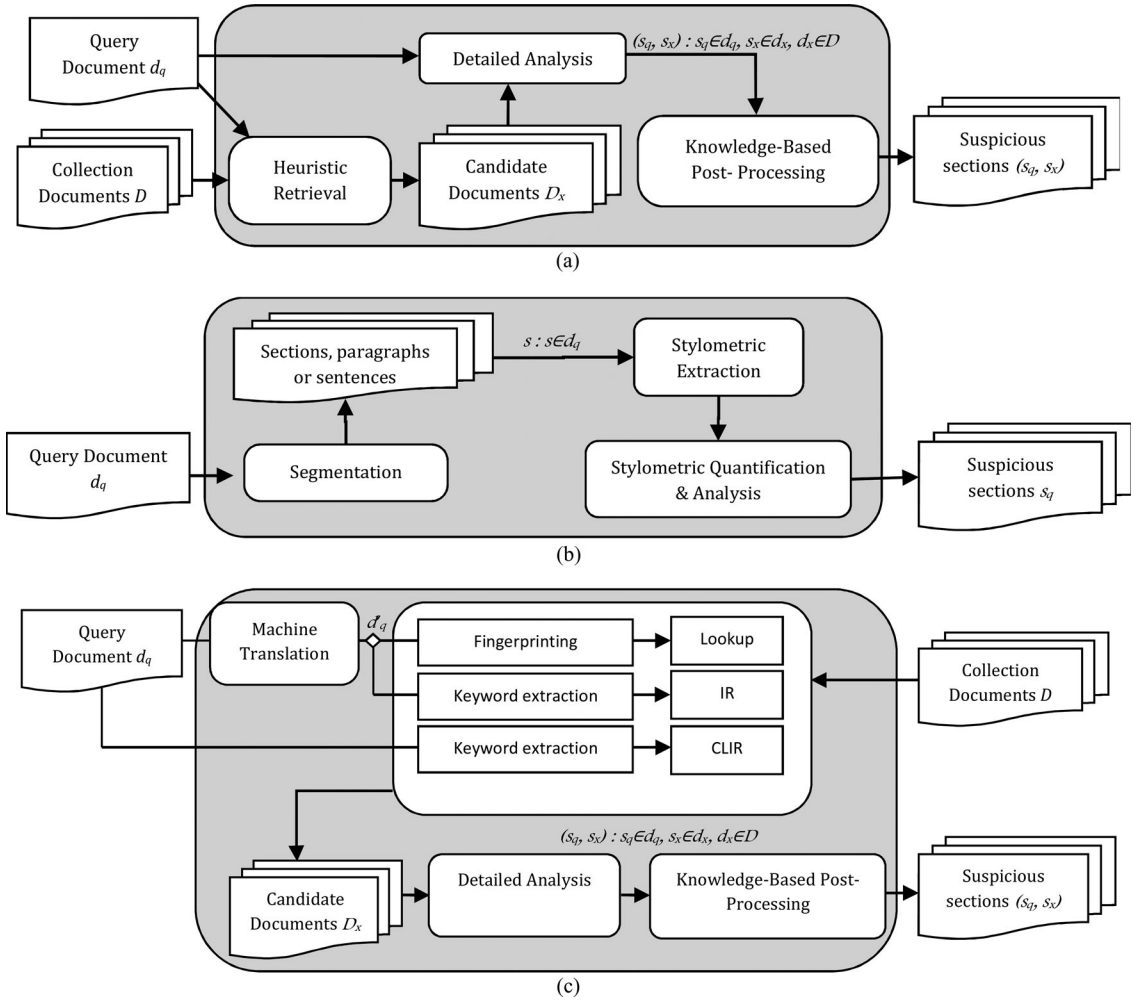

Fig. 10. Framework for different plagiarism detection systems. (a) White-box design for extrinsic plagiarism detection system [24], [25]. (b) White-box design for intrinsic plagiarism detection system. (c) White-box design for cross-lingual plagiarism detection system, inspired by [31].

the white-box design for intrinsic plagiarism detection. The operational framework can be summarized as follows [25]. The input is only a query document d_q (i.e., no reference collection D). Three main steps are needed (shown in rounded rectangles). First, a d_q is segmented into smaller parts, such as sections, paragraphs, or sentences. Text segmentation can be done on word level as well. Second, stylometric features, as will be explained in Section IV-B, are extracted from different segments. Third, stylometric-based measurements and quantification functions are employed to analyze the variance of different style features. Stylometry methods will be discussed later in Section V-B7. Parts with style which are inconsistent with the remaining document style are marked as possibly plagiarized and presented to humans for further investigation, i.e., the final output is fragments/sections s_q: $s_q \in d_q$ such that s_q has quantified writing style features different from other sections s in d_q.

D. Plagiarism Detection Languages

Plagiarism detection can be classified into monolingual and cross-lingual based on language homogeneity or heterogeneity of the textual documents being compared.

1) Monolingual Plagiarism Detection: Monolingual plagiarism detection deals with the automatic identification and extraction of plagiarism in a homogeneous language setting, e.g., English–English plagiarism. Most of the plagiarism detection systems have been developed for monolingual detection, which is divided into two former tasks, extrinsic and intrinsic, as discussed earlier.

2) Cross-Lingual Plagiarism Detection: Cross-language (or multilingual) plagiarism detection deals with the automatic identification and extraction of plagiarism in a multilingual setting, e.g., English–Arabic plagiarism. Research on cross-lingual plagiarism detection has attracted attention in recent few years [20], [27], [31], [36], [38], [108], thus focusing on text similarity computation across languages. Fig. 10(c) shows the white-box design for cross-lingual plagiarism detection. The operational framework can be summarized as follows [31]. The inputs are a query/suspicious document d_q in a language L_q, and a corpus collection D, such as the World Wide Web, which is expressed in multiple languages $L_1, L_2, \ldots, L_n$. Three steps are crucial (shown in rounded rectangles). First, a list of most promising documents D_x, where $D_x \in D$ is retrieved based on some CLIR model. Otherwise, D_x can be retrieved based on some IR model, if d_q is translated by using a machine translation

TABLE I
TYPES OF TEXT FEATURES WITH COMPUTATIONAL TOOLS REQUIRED FOR THEIR IMPLEMENTATION

-	Examples	Required Tools and Resources	Ref.
Lexical features	Character n-grams (fixed-length)	-	[1]
	Character n-grams (variable-length)	Feature selector (e.g. n-gram weights)	[16]
	Word n-grams	Tokenizer, [Stemmer, Lemmatizer]	[2, 3, 17, 26] [30]
Syntactic features	Chunks	Tokenizer, POS tagger, Text chunker (Windowing)	[4]
	Part-of-speech and phrase structure	Tokenizer, Sentence splitter, POS tagger	[6, 12, 48]
	Word position/order	Tokenizer, Sentence splitter, Compressor (e.g. Lempel-Zif)	[13, 14]
	Sentence	Tokenizer, Sentence splitter, POS tagger, Text chunker, Partial parser	[16, 58]
Semantic features	Synonyms, hyponyms, hypernyms, etc.	Tokenizer, [POS tagger], Thesaurus	[14, 16, 18, 58] [30]
	Semantic dependencies	Tokenizer, Sentence splitter, POS tagger, Text chunker, Partial parser, Semantic parser	[14, 61]
Structural features	Block-specific	HTML parser, Specialised parsers	[21, 29]
	Content-specific	Tokenizer, [Stemmer, Lemmatizer], Specialised dictionaries	-

Optional tools are included in brackets.

approach. Second, a pairwise feature-based detailed analysis is performed to find all suspicious parts s_q from d_q in language L_q that are similar to parts s_x from d_x, $d_x \in D_x$ in language L_x, where $L_q \neq L_x$. Finally, postprocessing operations are used to obtain the results in a human-readable format. Cross-language plagiarism detection task, therefore, contrasts extrinsic plagiarism detection task by bringing candidate set of documents D_x of different languages to be compared with the suspicious document d_q.

IV. TEXTUAL FEATURES

There are several textual features to quantify and characterize documents before applying a plagiarism detection method. This section discusses textual features needed in different frameworks: extrinsic, intrinsic, and cross-lingual.

A. Textual Features for Extrinsic Plagiarism Detection

Textual features to represent documents in extrinsic plagiarism detection include: *lexical* features, such as character *n*-gram and word *n*-gram; *syntactic* features, such as chunks, sentences, phrases, and POS, *semantic* features, such as synonyms and antonyms; and *structural* features that takes contextual information into account. Table I summarizes each types together with computational tools and resources required for their implementation. A detailed description of textual features for extrinsic plagiarism detection is given in the following.

1) Lexical Features: Lexical features operate at the character or word level. *Character-based n-gram (CNG)* representation is the simplest form whereby a document d is represented as a sequence of characters $d = \{(c_1, d), (c_2, d), \ldots, (c_n, d)\}$, where (c_i, d) refers to the ith character in d, and $n = d$ is the length of d (in characters). On the other hand, *word-based n-gram (WNG)* represents d as a collection of words $d = \{(w_1, d), (w_2, d), \ldots, (w_n, d)\}$, where (w_i, d) refers to the ith word in d, and $n = d$ is the length of d (in words) with ignoring sentence and structural bounds. Simple WNGs may be constructed by using *bigrams* (word-2-grams), *trigrams* (word-3-grams) or larger. *CNG* and *WNG* are commonly called *fingerprints* or *shingles* in text retrieval and plagiarism detection research. The process of generating fingerprints (or shingles) is called *fingerprinting* (or *shingling*). A document fingerprint can, therefore, identify the document uniquely as well as a human fingerprint does.

2) Syntactic Features: Syntactical features are manifested in part of speech (POS) of phrases and words in different statements. Basic POS tags include verbs, nouns, pronouns, adjectives, adverbs, prepositions, conjunctions, and interjections. POS tagging is the task of marking up the words in a text or more precisely in a statement as corresponding to a particular POS tag.

Sentence-based representation works by splitting the text into statements with the use of end-of-sentences delimiters, such as full stops, exclamation, and question marks. After splitting the text into sentences, POS and phrase structures can be constructed by using POS taggers. On the other hand, chunks is another feature that is generated by so-called windowing or sliding windows to characterize bigger text than phrases or sentences. POS could be further used in windowing to generate more expressive POS chunks. *Word order*, in a sentence or a chunk, could further be combined as a feature, and used as a comparison scheme between sentences.

3) Semantic Features: Semantic features quantify the use of word classes, synonyms, antonyms, hypernyms, and hyponyms. The use of thesaurus dictionaries and lexical databases, WordNet, for instance, would significantly provide more insights into the semantic meaning of the text. Together with POS tagging, semantic dependencies can be featured, and that would be very helpful in plagiarism detection.

4) Structural Features: Most plagiarism detection algorithms employ *flat* document features, such as lexical, syntactic, and semantic features. Very few algorithms have been developed to handle *structural* or *tree* features. Structural features reflect text organization and capture more document semantics. Documents can be described as a collection of paragraphs or passages, which can be considered as topical blocks. In many cases, paragraphs that are topically related or discuss the same subject can be grouped into sections, i.e., structural features might characterize documents as *headers*, *sections*, *subsections*, *paragraphs, sentences,* etc. This type of features can be used in structured documents, such as HTML webpages and XML files, and semistructured documents, such as books, theses, and academic journal papers. Note that structural features are most likely to be stored as XML trees for easier processing.

Structural features can be divided into *block-specific* and *content-specific*. In a recent study [29], *block-specific tree-structured features* were used to describe a collection of web

documents as blocks, namely, *document-page-paragraph*. Webpage documents were divided into paragraphs by an HTML parser taking the advantage of different HTML tags, such as <p>,<hr>, and
 to segment each webpage. Then, paragraphs were grouped into pages, whereby a new paragraph is added to each page until a maximum threshold of word count is reached; otherwise, a new page is created. Because paragraphs are more likely to have topically related sentences than pages, a recent study [21] encoded documents features in a hierarchical multilevel representation *document-paragraph-sentence*.

The existing structural implementations would be further improved, if the document features are encoded as *content-specific tree-structured features* by using semantically related blocks, such as *document-section-paragraph* or *class-concept-chunk*. The use of content-specific tree-structured features in combination with some *flat* features can be very useful in capturing the document's semantics and getting the gist of its sections/concepts. The rationale of using *content-specific* is to segment the document into different ideas (i.e., semantic blocks) to allow for the detection of idea plagiarism, in particular. Besides, we can drill down or roll up through the *structural* representation to detect more or less plagiarism types patterns, which are mentioned in our taxonomy of Fig. 2.

B. Textual Features for Intrinsic Plagiarism Detection

Stylometric features are based on the fact that each author develops an individual writing style. For example, authors employ, consciously or subconsciously, patterns to construct sentences, and use an individual vocabulary [33]. The stylometric features quantify various style aspects [22], [23], including 1) text statistics via various lexical features, which operate at the character or word level; 2) syntactic features, which work at the sentence level, quantify the use of word classes, and/or parse sentences into part of speech; 3) semantic features, which quantify the use of synonyms, functional words, and/or semantic dependencies; and 4) application-specific features, which reflect text organization, content-specific keywords, and/or other language-specific features. Table II summarizes the stylometric features together with computational tools and resources required for their implementation.

C. Textual Features for Cross-Lingual Plagiarism Detection

Features that are based on lexical and syntactic types are improper in a cross-lingual setting, i.e., for cross-lingual text relatedness and plagiarism detection, syntactic features are usually combined with semantic or statistical features. Other features may be language-specific or content-specific keywords. Table III summarizes textual features for cross-language plagiarism detection together with computational tools and resources required for their implementation.

V. Plagiarism Detection Methods

Different research works have described the methodology of *plagiarism detection* as stages (see Fig. 10). We will focus

TABLE II
Types of Stylometric Features With Computational Tools Required for Their Measurement [22], [23]

-	Examples	Tools and Resources	Ref.
Lexical features (Character-based)	Frequency of characters	-	[39]
	Character types (letters, digits, punctuations, etc.)	Character dictionaries	
	Frequency of special characters (e.g. !,& ,etc.)		
	Character n-grams (fixed-length) frequency	Chunker	[40, 41]
	Character n-grams (variable-length) frequency	Feature selector	
	Compression methods	Text compression tool	[22]
Lexical features (Word-based)	Token-based : - Average word length - Average sentence length - Average syllables per word	Tokenizer, [Sentence splitter]	[39, 49]
	Vocabulary richness - Type-token ratio (i.e. total unique vocabulary/total tokens) - Hapax legomena/dislegomena	Tokenizer	[39, 49-51]
	Frequency of words	Tokenizer, [Stemmer, Lemmatizer]	[40, 49]
	Frequency of function words	Tokenizer, Special dictionaries	[39, 40, 49, 56, 57]
	Word n-grams frequency	Tokenizer	[59]
	Averaged word frequency class	Tokenizer, [Stemmer, Lemmatizer]	[33]
	Lexical Errors - Spelling errors (e.g. letter omissions and insertions) - Formatting errors (e.g. all caps letters)	Tokenizer, Orthographic spell checker	[21, 57]
Syntactic features	Part-of-speech	Tokenizer, Sentence splitter, POS tagger	[22, 40]
	Part-of-speech n-gram frequency		[40, 57]
	Chunks	Tokenizer, Sentence splitter, [POS tagger],	[41, 66]
	Sentence and phrase structure	Tokenizer, Sentence splitter, POS tagger, Partial parser	[67]
	Rewrite rules frequencies	Tokenizer, Sentence splitter, POS tagger, Full parser	[68, 69]
	Syntactic Errors - Sentence fragments - Run-on sentences - Mismatched tense	Tokenizer, Sentence splitter, Syntactic spell checker	[57]
Semantic features	Synonyms, hypernyms, etc.	Tokenizer, [POS tagger], Thesaurus	[70]
	Semantic dependencies	Tokenizer, Sentence splitter, POS tagger, Partial parser, Semantic parser	[69]
	Functional	Tokenizer, Sentence splitter, POS tagger, Thesaurus, Specialised dictionaries	[72]
Application-specific	Structural - Average paragraph length - Indentation - Use of greetings and farewells - Use of signatures	HTML parser, Specialised parsers	[22, 39]
	Content-specific keywords	Tokenizer, [Stemmer, Lemmatizer], Specialised dictionaries	[39]
	Language-specific		[75, 76]

Optional tools are included in brackets.

TABLE III
TYPES OF CROSS-LANGUAGE TEXT FEATURES WITH COMPUTATIONAL TOOLS REQUIRED FOR THEIR IMPLEMENTATION

-	Examples	Required Tools and Resources	Ref.
Syntactic features	Word-n-grams (normally word-1-gram)	Tokenizer	[20]
	Chunks/fragments	Tokenizer, [Sentence splitter, POS tagger], Text chunker (Windowing)	[27]
	Word positions	Tokenizer, Sentence splitter, Compressor (e.g. Lempel-Zif)	[13, 20]
	Part-of-speech and phrase structure	Tokenizer, Sentence splitter, POS tagger	-
	Sentence	Tokenizer, Sentence splitter, POS tagger, Text chunker, Partial parser	[36]
Semantic features	Synonyms, hypernyms, etc.	Tokenizer, [POS tagger], Bilingual thesaurus	[20, 38]
	Semantic dependencies	Tokenizer, Sentence splitter, POS tagger, Text chunker, Partial parser, Semantic (bilingual) parser	-
Statistical features	Language-specific	Tokenizer, [Stemmer, Lemmatizer], Statistical (bilingual) dictionaries, Machine translators	[27, 36]

Optional tools are included in brackets.

TABLE IV
STRING SIMILARITY METRICS

Similarity Metric	Description	Examples	Ref.
Hamming distance (also edit distance)	defines number of characters different between two strings *x* and *y* of equal length. - allows only substitutions, cost 1	*x="aaabbbccc"* *y="aaabbbccd"* *d(x,y)=1*	-
Levenshtein distance (also edit distance)	defines minimum edit distance which transforms *x* into *y*. Edit operations include: -delete a char, cost 1 -insert a char, cost 1 -substitute one char for another, cost 1	*x="aaabbbccc"* *y="aaabbbccd"* *z="aaabbbccdf"* *w="aaabbbcc"* *d(x,y)=1* *d(x,z)=2* *d(x,w)=1*	[4, 5]
Longest Common Sequence (LCS) distance	measures the length of the longest pairing of chars that can be made between *x* and *y* with respect to the order of the chars. -allows insertions, cost 1 -allows deletions, cost 1	*x="aaabbbccc"* *y="aaabbbccd"* *d(x,y)=8*	[6, 28]

An example of similarity evaluation of different strings *x="aaabbbccc", y="aaabbbccd", z="aaabbbccdf" w="aaabbbcc".*

on two main stages. Section V-A is a step toward the stage, namely, *heuristic retrieval* of candidate documents. Section V-B is a step toward the stage, namely, *exhaustive analysis* of suspicious-candidate pairs and plagiarism detection.

A. Retrieval Models of Candidate Documents

Apart from intrinsic plagiarism detection, extrinsic plagiarism detection can be viewed as an IR task, and cross-lingual plagiarism detection can be seen as a CLIR task. Therefore, research on text plagiarism detection is merely analogous to text retrieval. In text retrieval, a list of documents are retrieved on the basis of the query, which could be small as few keywords or large as document size. Similarly, in plagiarism detection, a corpus collection of source documents D is searched to retrieve a list of *globally* similar documents to a *query document* d_q. With large datasets constructed recently [109], [110] and with the use of the Web as a corpus, research on plagiarism detection employs traditional IR models in order to retrieve a relatively small list of candidate documents to each suspicious document before applying a *locally* exhaustive analysis.

1) IR Models: Monolingual IR models have a broad theory and research [111]–[113]. Since the scope of this paper is beyond the theory of IR, we focus on the models applied as a prior step to plagiarism detection. Boolean model [114], the simplest, associates the query *terms* with Boolean operators (AND, OR, and NOT). A source collection, on the other hand, is represented as binary *vectors of terms*, and a term is either present, if it occurs at least once in the document representation, or not present, if it does not occur at all. In spite of its simplicity, Boolean model has no direct use in plagiarism detection because the query is usually of a document size.

Fingerprint [115], [116] and hash-based [117], [118] are common heuristic retrieval models, whereby source documents D and query document d_q are divided into small units, which are called fingerprints (or shingles) of length k. Fingerprinting (or shingling) technique is mentioned earlier (see Section IV). Hash-based model employs a hash function to transfer fingerprints, such as character or word k-grams, into hash values (or hash sums), which can be sorted and compared with other documents. The list of unique fingerprints (or their hashes) of each document is considered its vector. Vector similarity metrics (see Table V) can be used to retrieve documents, which share considerable number of fingerprints. Alzahrani and Salim [30] used word-3-gram fingerprints and retrieved candidate documents of d_q with Jaccard similarity above the threshold ($\alpha \geq 0.1$), Yerra and Ng [16], and Alzahrani [58] used three least-frequent character-4-gram and Jaccard similarity. Some research that has employed has retrieval include the following: using hashed word-5-gram fingerprints and Jaccard similarity [3] and using hashed fingerprints, where each fingerprint is 50-character chunks with 30-character overlap, and retrieval of documents that share at least one fingerprint with d_q [4].

Apart from earlier models, vector space model (VSM) [119] is also a very popular retrieval model that counts term frequency and inverse document frequency (IDF-TF) weighting scheme. The similarity between the weighted vectors of two documents is performed using one of the vector similarity metrics (see Table V). Research that has used VSM for candidate retrieval from large source collections include the following: using word-1-gram VSM and Cosine similarity [26], using word-8-gram VSM and custom distance measure [2], and using character-16-gram VSM and Cosine similarity [1].

Because of the lengthy term vectors of VSM, especially with large data collections, latent semantic indexing (LSI) [120] was developed for feature reduction while keeping the semantics of the document. LSI weighting scheme is based on the reduction of the original VSM (i.e., TF-IDF weighting vectors) by using singular value decomposition (SVD). LSI has been used to encode the semantics [121] and to widen the vocabulary by

TABLE V
VECTOR SIMILARITY METRICS

Vector Similarity Metric	Description & Equation	Equation	Range	Example	Ref.
Matching coefficient	-similar to Hamming distance but between vectors of equal length.	$M(x,y) = \lvert x \rvert - \lvert x \cap y \rvert$	0 to |x| Where |x|=|y|	*x=[0.1, 0.2, 0.3, 0.4] y=[0.1, 0.2, 0.3, 0.5] M(x,y)= 1*	[11]
Jaccard (or Tanimoto) coefficient	-defines number of shared terms against total number of terms. This measure is computed to one if vectors are identical.	$J(x,y) = \frac{\lvert x \cap y \rvert}{\lvert x \cup y \rvert}$	0 to 1	*J(x,y)=3/5=0.6*	[3, 7, 8, 21]
Dice's coefficient	-similar to Jaccard but reduces the effect of shared terms between vectors. This measure is computed to two if vectors are identical.	$D(x,y) = \frac{2 \lvert x \cap y \rvert}{\lvert x \cup y \rvert}$	0 to 2	*D(x,y)=6/5=1.2*	-
Overlap (or containment) coefficient	-if v_1 is subset of v_2 or the converse, then the similarity coefficient is a full match.	$O(x,y) = \frac{\lvert x \cap y \rvert}{\min(\lvert x \rvert, \lvert y \rvert)}$	0 to 1	*O(x,y)=3/4=0.75 (or 75%)*	[10]
Cosine coefficient	-finds the cosine angle between two vectors.	$Cos(x,y) = \frac{\sum_i (x_i, y_i)}{\sqrt{\sum_i (x_i)^2} \sqrt{\sum_i (y_i)^2}}$	0 to 1	*Cos(x,y)=0.34/0.3421 =0.9939 ≈1*	[9, 21, 26, 28]
Euclidean distance	-measures the geometric distance between two vectors.	$Ec(x,y) = \sqrt{\sum_i \lvert x_i - y_i \rvert^2}$	0 to ∞	*Ec(x,y)=0.1*	-
Squared Euclidean Distance	-places progressively greater weight on vectors that are further apart	$SEc(x,y) = \sum_i (x_i - y_i)^2$	0 to ∞	*SEc(x,y)=0.01*	-
Manhattan Distance	-measures the average difference across dimensions and yields results similar to the simple Euclidean distance	$Manh(x,y) = \sum_i \lvert x_i - y_i \rvert^2$	0 to ∞	*Manh(x,y)=0.1*	-

An example of similarity evaluation of two vectors *x= [0.1, 0.2, 0.3, 0.4]* and *y= [0.1, 0.2, 0.3, 0.5]*.

incorporating thesaurus dictionaries [122]. The LSI model has been applied for candidate retrieval and plagiarism detection in [88] and [123].

VSM and LSI only consider the global similarity of documents which may not lead to the detection of plagiarism. In other words, many documents that are globally similar may not contain plagiarized paragraphs or sentences. Zhang and Chow [21], therefore, proposed the incorporation of structural features (*document-paragraph-sentence*) into the candidate retrieval stage. Two retrieval methods were used: histogram-based multilevel matching (MLMH) and signature-based multilevel matching (MLMS). In MLMH, global similarity at the document level and local similarity at the paragraph level are hybridized into a single measure, where each similarity is obtained by matching word histograms of their representatives (i.e., document or paragraph). MLMS adds a weight parameter to the word histograms in order to consider the so-called the information capacity, or the proportion of words in the histogram vector against the total words in its representative.

Fuzzy retrieval [124]–[126] has become popular and was mainly developed to generalize the Boolean model by considering a partial relevance between the query and the data collection. Fuzzy set theories deal with the representation of classes whose boundaries are not well defined [127]. Each element of the class is associated with a membership function that defines the membership degree of the element in the class. In [16], fuzzy set IR model was applied to retrieve documents that share similar, but not necessarily same, statements above a threshold value. In many fuzzy representation approaches, the TF-IDF function of the weighted vector model is used as the fuzzy membership function.

Some models, such as language model [128] that ranks documents by using statistical methods, and probabilistic model [129] that assigns relevance scores to terms and uses probabilistic methods for document ranking, have yet to be applied for candidate retrieval and plagiarism detection.

2) Clustering Techniques: Clustering is concerned with grouping together documents that are similar to each other and dissimilar to documents belonging to other clusters. There are many algorithms for clustering that use a distance measure between clusters, including flat and hierarchical clustering [130], [131] and may also account the user's viewpoint during clusters construction [132]. Self-organizing map (SOM) is a form of unsupervised neural networks that are introduced by Kohonen [133] and exhibits interesting features of a data collection, such as self-organizing and competitive learning. SOM was used for features projection, document clustering, and cluster visualization [134], [135]. In [136], WEBSOM investigated the use of SOM in clustering and classifying large collections on the basis of statistical word histograms, and was able to reduce high-dimensional features to 2-D maps. In [137], LSISOM investigated the use of SOM in clustering a document collection by encoding the LSI of document terms rather than statistical word category histograms in WEBSOM. Clustering techniques are not enough by itself to judge plagiarism but can be used in the candidate retrieval stage to group similar documents that discuss the same subject. It should be followed by another level of plagiarism analysis and detection methods. In [138], a plagiarism detection system used clustering to find similar documents; then, documents in the same cluster were compared until two similar paragraphs are found. Paragraphs were compared in detail, i.e., on a sentence-per-sentence basis to highlight plagiarism. In [29], a method that uses multilayer SOM (ML-SOM) was developed for effective candidate retrieval of a set of similar documents for a suspected document d_q and plagiarism detection. In the aforementioned ML-SOM, the top layer performs document clustering and retrieval, and the bottom layer plays

TABLE VI
PLAGIARISM DETECTION METHODS AND THEIR EFFICIENCY IN DETECTING DIFFERENT PLAGIARISM TYPES

Technique	Tasks		IR		Language(s)	Plagiarism Type(s)								Reference
						Literal			Intelligent					
	extrinsic	intrinsic	mono-lingual	cross-lingual		copy	near copy	restructuring	paraphrasing	summarising	translating	idea (section)	idea (context)	
Char-Based (CNG)	☑		☑		any	☑	☑							[1-6]
Vector-Based (VEC)	☑		☑		any	☑	☑	☑						[7-11]
Syntax-Based (SYN)	☑		☑		specific	☑	☑	☑						[6, 12, 13]
Semantic-Based (SEM)	☑		☑		specific	☑	☑	☑	☑	☐				[14, 15]
Fuzzy-Based (FUZZY)	☑		☑		specific	☑	☑	☑	☑	☐				[16-19]
Structural-Based (STRUC)	☑		☑		specific	☑	☑	☑	☐	☐		☐	☐	[21, 29]
Stylometric-Based (STYLE)		☑	☑		specific	☑	☑	☑						[22, 23, 32-35]
Cross-Lingual (CROSS)	☑			☑	cross						☑			[31, 36-38]

The notions in the table indicate the following: ☑ means include/support by evidence from research stated in the references column, ☐ means possibility to include/support but need further research for proof.

an important role in detecting similar, potentially plagiarized, paragraphs.

3) Cross-Lingual Retrieval Models: Fig. 10(c) shows three alternatives for candidate retrieval in a cross-lingual plagiarism detection task [31]. The first one is cross-lingual information retrieval (CLIR) whereby documents that are globally similar to the suspicious document are retrieved in a multilingual stetting. Keywords from the query document d_q are extracted and then translated to the language of source collection and then used for querying the index words of source collection D, as in normal CLIR models.

Besides CLIR, the second alternative is monolingual IR with machine translation, where by d_q is translated by using a machine translation algorithm then followed by normal IR methods for retrieval. Third is hash-based search, where the translation of d_q is fingerprinted and then hashed using hashing functions. A similarity hash function is used for querying and retrieving the fingerprint hash index of documents in D.

B. Exhaustive Analysis Methods of Plagiarism Detection

Methods to compare, manipulate, and evaluate textual features in order to find plagiarism can be categorized into eight types: CNG, VEC, SYN, SEM, FUZZY, STRUC, STYLE, and CROSS. Subsequent sections will describe each category in detail.

1) Character-Based Methods: The majority of plagiarism detection algorithms rely on character-based lexical features, word-based lexical features, and syntax features, such as sentences, to compare the query document d_q with each candidate document $d_x \in D_x$. Matching strings in this context can be *exact* or *approximate*. *Exact string matching* between two strings x and y means that they have exactly same characters in the same order. For example, the character 8-gram string $x =$ "*aaabbbcc*" is exactly the same as *"aaabbbcc"* but differs from $y =$ "*aaabbbcd*."

Different plagiarism techniques, which feature the text as character n-gram or word n-gram, use exact string matching. For instance, Grozea *et al*. [1] used character 16-gram matching, [2] word 8-gram matching, and [3] word 5-gram matching.

On the other hand, *approximate string matching* shows, to some degree, that two strings x and y are similar/dissimilar. For instance, the character 9-gram $x =$ "*aaabbbccc*" and $y =$ "*aaabbbccd*" are highly similar because all letters match except the last one. Numerous metrics measure the *distance* between strings in different ways. The distance $d(x, y)$ between two strings x and y is defined as follows:

> "The minimal cost of a sequence of operations that transform x into y and the cost of a sequence of operations is the sum of the costs of the individual operations" [139].

Possible *operations* that could transfer one string into another are [139], [140]

1) *Insertion* (s, a): inserting letter a into string s;
2) *Deletion* (a, s): deleting letter a from string s;
3) *Substitution or replacement* (a, b): substituting letter a by b;
4) *Transposition* (ab, ba): swapping adjacent letters a and b.

Examples of string similarity metrics include *hamming* distance, which allows only substitutions at cost 1, *levenshtein* distance, which allows insertions, deletions, and substitutions at cost 1, and *longest common subsequence (LCS)* distance, which allows only insertions and deletions at cost 1. Table IV summarizes the description of each metric and gives examples. The table also refers to some research that applied these metrics in plagiarism detection.

Approximate string matching and similarity metrics have been widely used in plagiarism detection. Scherbinin and Butakov [4] used *Levenshtein* distance to compare word n-gram and combine adjacent similar grams into sections. Su *et al*. [5] combined *Levenshtein* distance, and simplified *Smith-Waterman* algorithm for the identification and quantification of local similarities in plagiarism detection. Elhadi and Al-Tobi [6] used the *LCS* distance combined with other POS syntactical features to identify similar strings locally and rank documents globally.

2) Vector-Based Methods: Lexical and syntax features may be compared as vectors of terms/tokens rather than strings. The similarity can be computed by using vector similarity coefficients [140], i.e., word *n*-gram is represented as a vector of *n* terms/tokens, sentences and chunks are resented as either term vectors or character *n*-grams vectors; then, the similarity can be evaluated by using matching, Jaccard (or Tanimoto), Dice's, overlap (or containment), Cosine, Euclidean, or Manhattan coefficients. Table V describes these vector similarity metrics with mathematical representation and supporting example.

A number of research works on plagiarism detection have mainly used Cosine and Jaccard. Murugesan *et al*. [9] used Cosine similarity on the entire document or on document fragments to enable the global or partial detection of plagiarism without sharing the documents' content. Due to its simplicity, the use of Cosine with other similarity metrics was efficient for plagiarism detection in secured systems in which submissions are considered confidential, such as conferences. Zhang and Chow [21] used exponential Cosine distance as a measure of document dissimilarity that globally converges to 0 for small distances and to 1 for large distances. To encompass statements that are locally similar to the final decision of plagiarism detection, Jaccard coefficient was used to estimate the overlap between sentences. Barrón-Cedeño *et al*. [7] estimated the similarity between *n*-gram terms of different lengths $n = \{1,2, \ldots, 6\}$ by using Jaccard coefficient. Similarly, Lyon *et al*. [8] exploited the use of word trigrams to measure the similarity of short passages in large document collections.

On the other hand, Barrón-Cedeño and Rosso [10] used the containment metric to compare chunks from documents, which was based on word *n*-gram, $n = \{2,3\}$. The resulting vectors of word *n*-grams and containment similarity were used to show the degree of overlapping between two fragments. Daniel and Mike [11] used the matching coefficient with a threshold to score similar statements.

3) Syntax-Based Methods: Some research works have used syntactical features to gauge the text similarity and plagiarism detection. In recent studies, Elhadi and Al-Tobi [6] and Elhadi and Al-Tobi [12] used POS tags features followed by other string similarity metrics in the analysis and calculation of similarity between texts. This is based on the intuition that similar (exact copies) documents would have similar (exact) syntactical structure (sequence of POS Tags). The more POS tags are used, the more reliable features are produced to measure similarity, i.e., similar documents and, in particular, those that contain some exact or near-exact parts of other documents would contain similar syntactical structures.

Elhadi and Al-Tobi [12] proposed an approach that looks at the use of syntactical POS tags to represent text structure as a basis for further comparison and analysis. POS tags were refined and used for document ranking. Similar documents, in terms of POS features, were carried for further analysis and for presenting sources of plagiarism. The previous methodology was also used in [6], but strings were compared by using a refined and improved LCS algorithm in the matching and ranking phase.

Cebrián *et al*. [13] used Lempel–Ziv algorithm to compress the syntax and morphology of two texts based on a normalized distance measure and compare shared topological information that is given by the compressor. The method was able to detect similar texts, even if they have different literals.

In contrast to many existing plagiarism detection systems that reduce the text into a set of tokens by removing stop words and stemming, the approach in [6], [12], and [13] reduces the text into a smaller set of syntactical tags, thus making use of most of the content.

4) Semantic-Based Methods: A sentence can be treated as a group of words arranged in a particular order. Two sentences can be semantically the same but differ in their structure, e.g., by using the active versus passive voice, or differing in their word choice. Semantic approaches seem to have had less attention in plagiarism detection, which could be due to the difficulties of representing semantics, and the complexities of representative algorithms. Li *et al*. [14] and Bao *et al*. [15] used semantic features for similarity analysis and obfuscated plagiarism detection.

In [14], a method to calculate the semantic similarity between short passages of sentence length is proposed based on the information extracted from a structured lexical database and corpus statistics. The similarity of two sentences is derived from *word similarity* and *order similarity*. The word vectors for two pairs of sentences are obtained by using unique terms in both sentences and their synonyms from WordNet, besides term weighting in the corpus. The order similarity defines that different word order may convey different meaning and should be counted into the total string similarity.

In [15], a so-called semantic sequence kin (SSK) is proposed based on the *local semantic density*, not on the common global word frequency. SSK first finds out the *semantic sequences* based on the concept of *semantic density*, which represents locally the frequent semantic features, and then all of the semantic sequences are collected to imply the global features of the document. In spite of its complexity, this method was found to be ideal in detecting reworded sentences, which greatly improves the precision of the results.

5) Fuzzy-Based Methods: In fuzzy-based methods, matching fragments of text, such as sentences, become approximate or vague, and implements a spectrum of similarity values that range from one (exactly matched) to zero (entirely different).

The concept "fuzzy" in plagiarism detection can be modeled by considering that each word in a document is associated with a fuzzy set that contains words with same meaning, and there is a degree of similarity between words in a document and the fuzzy set [16]. In a statement-based plagiarism detection, fuzzy approach was found to be effective [16], [18], [19] because it can detect similar, yet not necessarily the same, statements based on the similarity degree between words in the statements and the fuzzy set. The question is how to construct the fuzzy set and the degree of similarity between words.

In [16], a *term-to-term correlation matrix* is constructed, which consists of words and their corresponding *correlation factors* that measure the degrees of similarity (degree of membership between 0 and 1) among different words, such as "automobile" and "car." Then, the degree of similarity among sentences can be obtained by computing the correlation factors

between any pair of words from two different sentences in their respective documents. The *term-to-term correlation factor* $F_{i,j}$ defines a fuzzy similarity between two words w_i and w_j as follows:

$$F_{i,j} = \frac{n_{i,j}}{n_i + n_j - n_{i,j}}$$

where $n_{i,j}$ is the number of documents in a collection with both words w_i and w_j, and n_i (n_j, respectively) is the number of documents with w_i (w_j, respectively) in the collection.

In [30], the term-to-term correlation factor was replaced by a fuzzy similarity, which is defined as follows:

$$F_{i,j} = \begin{cases} 1.0, & \text{if} \quad w_i = w_j \\ 0.5, & \text{if} \quad w_i = \text{synset}(w_j) \\ 0.0, & \text{otherwise.} \end{cases}$$

The synset of the word w_i was extracted by using WordNet semantic web database [141]. Then, the degree of similarity of two sentences is the extent to which sentences are matched. To obtain the degree of similarity between two sentences S_i and S_j, we first compute the *word–sentence correlation factor* $\mu_{i,j}$ of w_i in S_i with all words in S_j , which measures the degree of similarity between w_i and (all the words in) S_j, as follows [16]:

$$\mu_{i,j} = 1 - \prod\nolimits_{w_k \in S_j} (1 - F_{i,k})$$

where w_k is every word in S_j, and $F_{i,k}$ is the correlation factor between w_i and w_k.

Based on the μ-value of each word in a sentence S_i, which is computed against sentence S_j, the *degree of similarity* of S_i with respect to S_j can be defined as follows [16]:

$$\text{Sim}(S_i, S_j) = \frac{(\mu_{1,j} + \mu_{2,j} + \cdots + \mu_{n,j})}{n}$$

where w_k ($1 \leq k \leq n$) is a word in S_i, and n is the total number of words in S_i. Sim(S_i,S_j) is a normalized value. Likewise, Sim(S_j,S_i), which is the *degree of similarity* of S_j with respect to S_i, is defined accordingly.

Using the equation as defined earlier, two sentences S_i and S_j should be treated the same, i.e., equal (EQ), according to the following equation [16]:

$$\text{EQ}(S_i, S_j) = \begin{cases} 1 \quad \text{if } \min(\text{Sim}(S_i, S_j), \text{Sim}(S_j, S_i)) \geq p \wedge \\ \qquad |\text{Sim}(S_i, S_j) - \text{Sim}(S_j, S_i)| \leq v \\ 0, \quad \text{otherwise} \end{cases}$$

where p, which is the *permission threshold value*, was set to 0.825, whereas v, which is the *variation threshold value,* was set to 0.15 [16]. Permission threshold is the minimal similarity between any two sentences S_i and S_j, which is used partially to determine whether S_i and S_j should be treated as equal (*EQ*). On the other hand, variation threshold value is used to decrease false positives (statements that are treated as equal but they are not) and false negatives (statements that are equal but treated as different) [16].

Subsequent to the previous work, Koberstein and Ng [17] developed a reliable tool by using fuzzy IR approach to determine the degree of similarity between any two web documents and clustering web documents with similar, but not necessarily the same, content. In addition, Alzahrani and Salim [18], [19] adapted the fuzzy IR model for use with Arabic scripts by using a plagiarism corpus of 4477 source statements and 303 query/suspicious statements. Experimental results showed that fuzzy IR can find to what extent two Arabic statements are similar or dissimilar.

6) Structural-Based Methods: It is worth noting that all the aforementioned methods use *flat* features representation. In fact, flat feature representations use lexical, syntactic, and semantic features of the text in the document, but do not take into account contextual similarity, which are based on the ways the words are used throughout the document, i.e., sections and paragraphs. Moreover, until now, most document models incorporate only term frequency and do not include such contextual information. *Tree-structured* features representation is a rich data characterization, and ML-SOM are very effective in handling such contextual information [134], [142]. Chow and Rahman [29] made a quantum leap to use *block-specific* tree-structured representation and to utilize ML-SOM for plagiarism detection. The top layer performs document clustering and candidate retrieval, and the bottom layer plays an important role in detecting similar, potentially plagiarized, paragraphs by using Cosine similarity coefficient.

7) Stylometric-Based Methods: Based on stylometric features, formulas can be constructed to quantify the characteristics of the writing style. Research on intrinsic plagiarism detection [32]–[35] has focused on quantifying the trend (or complexity) of style that a document has. Style-quantifying formulas can be classified according to their intention: *writer-specific* and *reader-specific* [33]. Writer-specific formulas aim to quantify the author's vocabulary richness and style complexity. Reader-specific formulas aim to grade the level that is required to understand a text. Recent research areas include outlier analysis, metalearning, and symbolic knowledge processing, i.e., knowledge representation, deduction, and heuristic inference [23]. Stamatatos [22], and Stein *et al*. [23] excellently reviewed the state-of-the-art stylometric-based methods until 2009 and 2010, respectively, and many details can be found within.

8) Methods for Cross-Lingual Plagiarism Detection: Detailed analysis methods of cross-language plagiarism detection were surveyed in [31]. Cross-lingual methods are based on the measurement of the similarity between sections of the suspicious document d_q and sections of the candidate document in d_x based on cross-language text features. Methods include 1) cross-lingual syntax-based methods, which use character *n*-grams features for languages that are syntactically similar, such as European languages [31]; 2) cross-lingual dictionary-based methods [38]; 3) cross-lingual semantic-based methods, which use comparable or alignment corpora that exploits the vocabulary correlations [31], [37]; and 4) statistics-based methods [36].

VI. Plagiarism Types, Features and Methods: Which Method Detects Which Plagiarism?

The taxonomy of plagiarism (see Fig. 2) illustrates different types of plagiarism on the basis of the way the offender (purposely) changes the plagiarized text. Plagiarism is categorized into literal plagiarism (refers to copying the text nearly as it is)

and intelligent plagiarism (refers to illegal practices of changing texts to hide the offence including restructuring, paraphrasing, summarizing, and translating). Adoption of (embracing as your own) ideas of other is a type of intelligent plagiarism, where a plagiarist deliberately 1) chooses texts that convey creative ideas, contributions, results, findings, and methods of solving problems; 2) obfuscates how these ideas were written; and 3) embeds them within another work without giving credit to the source of ideas. We categorize idea plagiarism, based on its occurrence within the document, into three levels: the lowest is the *semantic-based meaning* at the paragraph (or sentence) level, the intermediate is the *section-based importance* at the section level, and the top (or holistic) is the *context-based adaptation*, which is based on ideas structure in the document.

Textual features are essential to capture different types of plagiarism. Implementing rich feature structures should lead to the detection of more types of plagiarism, if a proper method and similarity measure are used as well. *Flat-feature* extraction includes *lexical*, *syntactic,* and *semantic* features, but does not account contextual information of the document. *Structural-feature* (or *tree-feature*) extraction, on the other hand, takes into account the way words are distributed throughout the document. We categorize structural features into block-specific [21], [29], which encodes the document as hierarchical blocks (*document-page-paragraph* or *document-paragraph-sentence*), and content-specific, which encodes the content as semantic-related structure (*document-section-paragraph* or *class-concept-chunk*). The latter, combined with *flat* features, is suitable to capture a document's semantics and get the gist of its concepts. Besides, we can drill down or roll up through the *tree* representation to detect more plagiarism patterns.

Many plagiarism detection methods focus on copying text with/without minor modification of the words and grammar. In fact, most of the existing systems fail to detect plagiarism by paraphrasing the text, by summarizing the text but retaining the same idea, or by stealing ideas and contributions of others. This is why most of the current methods do not account the overlap when a plagiarized text is presented in different words. Table VI compares and contrasts differences between various techniques in detecting different types of plagiarism, which are stated in the taxonomy (see Fig. 2).

To illustrate, we discuss the reliability and efficiency, in general, with pointing out some pros and cons of each method. CNG and VEC methods [1]–[11] are performed by dividing documents into small nontopical blocks, such as words or characters *n*-grams. SYN methods [6], [12] are based on decomposing documents into statements and extracting POS features. Plagiarism detection process in CNG, VEC, and SYN is sped up but at the expense of losing semantic information. Therefore, they are unable types other than literal plagiarism.

On the other hand, SEM [14], [15] and FUZZY [16]–[19] methods incorporate different semantic features, thesaurus dictionaries, and lexical databases, such as WordNet. They are more reliable than earlier methods because they can detect plagiarism by rewording the words and rephrasing the content. However, the SEM and FUZZY approaches seem to have received less attention in plagiarism detection research due to the challenge of representing semantics and the time complexity of representative algorithms, which make them inefficient and impractical for real-world tools.

STYLE [22], [23], [32]–[35] methods are meant to analyze the writing style and suspect plagiarism within a document, in the absence of using a source collection for comparison. Changes and variations in the writing style may indicate that the text is copied (or nearly copied) from another author. However, an evidence of plagiarism (i.e., providing the source document) is crucial to a human investigator, a disadvantage of STYLE methods.

Unlike earlier methods, STRUC methods [21], [29] use contextual information (i.e., *topical* blocks, sections, and paragraphs), which carries different importance of text, and characterizes different ideas distributed throughout the document. STRCU integrated with VEC methods have been used to detect copy-and-paste plagiarism [21], [29], but further research should be carried out to investigate the advantages of relating STRUC with SEM and FUZZY methods for idea plagiarism detection.

VII. Conclusion

Current antiplagiarism tools for educational institutions, academicians, and publishers "can pinpoint only word-for-word plagiarism and only some instances of it" [65] and do not cater adopting ideas of others [65]. In fact, idea plagiarism is awfully more successful in the academic world than other types because academicians may not have sufficient time to track their own ideas, and publishers may not be well-equipped to check where the contributions and results come from [73]. As plagiarists become increasingly more sophisticated, idea plagiarism is a key academic problem and should be addressed in future research. We suggest that the SEM and FUZZY methods are proper to detect *semantic-based meaning* idea plagiarism at the paragraph level, e.g., when the idea is summarized and presented in different words. We also propose the use of structural features and contextual information with efficient STRUC-based methods to detect *section-based importance* and *context-based adaptation* idea plagiarism.

References

[1] C. Grozea, C. Gehl, and M. Popescu, "ENCOPLOT: Pairwise sequence matching in linear time applied to plagiarism detection," in *Proc. SEPLN*, Donostia, Spain, 2012, pp. 10–18.
[2] C. Basile, D. Benedetto, E. Caglioti, G. Cristadoro, and M. D. Esposti, "A plagiarism detection procedure in three steps: Selection, matches and "squares," in *Proc. SEPLN*, Donostia, Spain, pp. 19–23.
[3] J. Kasprzak, M. Brandejs, and M. Křipač, "Finding plagiarism by evaluating document similarities," in *Proc. SEPLN*, Donostia, Spain, pp. 24–28.
[4] V. Scherbinin and S. Butakov, "Using Microsoft SQL server platform for plagiarism detection," in *Proc. SEPLN*, Donostia, Spain, pp. 36–37.
[5] Z. Su, B. R. Ahn, K. Y. Eom, M. K. Kang, J. P. Kim, and M. K. Kim, "Plagiarism detection using the Levenshtein distance and Smith-Waterman algorithm," in *Proc. 3rd Int. Conf. Innov. Comput. Inf. Control*, Dalian, Liaoning, China, Jun. 2008, p. 569.
[6] M. Elhadi and A. Al-Tobi, "Duplicate detection in documents and webpages using improved longest common subsequence and documents syntactical structures," in *Proc. 4th Int. Conf. Comput. Sci. Converg. Inf. Technol.*, Seoul, Korea, Nov. 2009, pp. 679–684.

[7] A. Barrón-Cedeño, C. Basile, M. Degli Esposti, and P. Rosso, "Word length n-Grams for text re-use detection," in *Computational Linguistics and Intelligent Text Processing*, 2010, pp. 687–699.

[8] C. Lyon, J. A. Malcolm, and R. G. Dickerson, "Detecting short passages of similar text in large document collections," in *Proc. Conf. Emp. Methods Nat. Lang. Process.*, 2001, pp. 118–125.

[9] M. Murugesan, W. Jiang, C. Clifton, L. Si, and J. Vaidya, "Efficient privacy-preserving similar document detection," *VLDB J.*, vol. 19, no. 4, pp. 457–475, 2010.

[10] A. Barrón-Cedeño and P. Rosso, "On automatic plagiarism detection based on n-grams comparison," in *Proc. 31st Eur. Conf. IR Res. Adv. Info. Retrieval*, 2009, pp. 696–700.

[11] R. W. Daniel and S. J. Mike, "Sentence-based natural language plagiarism detection," *ACM J. Edu. Resour. Comput.*, vol. 4, p. 2, 2004.

[12] M. Elhadi and A. Al-Tobi, "Use of text syntactical structures in detection of document duplicates," in *Proc. 3rd Int. Conf. Digital Inf. Manage.*, London, U.K., 2008, pp. 520–525.

[13] K. Koroutchev and M. Cebrián, "Detecting translations of the same text and data with common source," *J. Stat. Mech.: Theor. Exp.*, p. P10009, 2006.

[14] Y. Li, D. McLean, Z. A. Bandar, J. D. O'Shea, and K. Crockett, "Sentence similarity based on semantic nets and corpus statistics," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 8, pp. 1138–1150, Aug. 2006.

[15] J. P. Bao, J. Y. Shen, X. D. Liu, H. Y. Liu, and X. D. Zhang, "Semantic sequence kin: A method of document copy detection," in *Advances in Knowledge Discovery and Data Mining*, 2004, pp. 529–538.

[16] R. Yerra and Y.-K. Ng, "A sentence-based copy detection approach for web documents," in *Fuzzy System and Knowledge Discovery*, 2005, pp. 557–570.

[17] J. Koberstein and Y.-K. Ng, "Using word clusters to detect similar web documents," in *Knowledge Science, Engineering and Management*, 2006, pp. 215–228.

[18] S. Alzahrani and N. Salim, "On the use of fuzzy information retrieval for gauging similarity of arabic documents," in *Proc. 2nd Int. Conf. Appl. Digital Inf. Web Technol.*, 2009, pp. 539–544.

[19] S. Alzahrani and N. Salim, "Statement-based fuzzy-set IR versus fingerprints matching for plagiarism detection in arabic documents," in *Proc. 5th Postgraduate Annu. Res. Seminar*, Johor Bahru, Malaysia, 2009, pp. 267–268.

[20] Z. Ceska, M. Toman, and K. Jezek, "Multilingual plagiarism detection," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence Lecture Notes in Bioinformatics)*, vol. 5253 LNAI, pp. 83–92, 2008.

[21] H. Zhang and T. W. S. Chow, "A coarse-to-fine framework to efficiently thwart plagiarism," *Pattern Recog.*, vol. 44, pp. 471–487, 2011.

[22] E. Stamatatos, "A survey of modern authorship attribution methods," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 60, pp. 538–556, 2009.

[23] B. Stein, N. Lipka, and P. Prettenhofer, "Intrinsic plagiarism analysis," in *Language Resources & Evaluation*, 2010.

[24] B. Stein, S. M. z. Eissen, and M. Potthast, "Strategies for retrieving plagiarized documents," in *Proc. 30th Annu. Int. ACM SIGIR*, Amsterdam, The Netherlands, 2007, pp. 825–826.

[25] M. Potthast, B. Stein, A. Eiselt, A. Barrón-Cedeño, and P. Rosso, "Overview of the 1st international competition on plagiarism detection," in *Proc. SEPLN*, Donostia, Spain, pp. 1–9.

[26] M. Zechner, M. Muhr, R. Kern, and M. Granitzer, "External and intrinsic plagiarism detection using vector space models," in *Proc. SEPLN*, Donostia, Spain, pp. 47–55.

[27] A. Barrón-Cedeño, P. Rosso, D. Pinto, and A. Juan, "On cross-lingual plagiarism analysis using a statistical model," in *Proc. ECAI PAN Workshop*, Patras, Greece, pp. 9–13.

[28] A. Parker and J. O. Hamblen, "Computer algorithms for plagiarism detection," *IEEE Trans. Educ.*, vol. 32, no. 2, pp. 94–99, May 1989.

[29] T. W. S. Chow and M. K. M. Rahman, "Multilayer SOM with tree-structured data for efficient document retrieval and plagiarism detection," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1385–1402, Sep. 2009.

[30] S. Alzahrani and N. Salim, "Fuzzy semantic-based string similarity for extrinsic plagiarism detection: Lab report for PAN at CLEF'10," presented at the 4th Int. Workshop PAN-10, Padua, Italy, 2010.

[31] M. Potthast, A. Barrón-Cedeño, B. Stein, and P. Rosso, "Cross-language plagiarism detection," *Language Resources & Evaluation*, pp. 1–18, 2010.

[32] G. Stefan and N. Stuart, "Tool support for plagiarism detection in text documents," in *Proc. ACM Symp. Appl. Comput.*, Santa Fe, NM, 2005, pp. 776–781.

[33] M. zu Eissen, B. Stein, and M. Kulig, "Plagiarism detection without reference collections," in *Advances in Data Analysis*, 2007, pp. 359–366.

[34] S. zu Eissen and B. Stein, "Intrinsic plagiarism detection," in *Advances in Information Retrieval*, 2006, pp. 565–569.

[35] A. Byung-Ryul, K. Heon, and K. Moon-Hyun, "An application of detecting plagiarism using dynamic incremental comparison method," in *Proc. Int. Conf. Comput. Intell. Security*, Guangzhou, China, 2006, pp. 864–867.

[36] D. Pinto, J. Civera, A. Barrón-Cedeño, A. Juan, and P. Rosso, "A statistical approach to crosslingual natural language tasks," *J. Algorithms*, vol. 64, pp. 51–60, 2009.

[37] M. Potthast, B. Stein, and M. Anderka, "A Wikipedia-based multilingual retrieval model," in *Advances in Information Retrieval*, 2008, pp. 522–530.

[38] R. Corezola Pereira, V. Moreira, and R. Galante, "A new approach for cross-language plagiarism analysis," in *Multilingual and Multimodal Information Access Evaluation,* vol. 6360, M. Agosti, N. Ferro, C. Peters, M. de Rijke, and A. Smeaton, Eds. Berlin, Germany: Springer, 2010, pp. 15–26.

[39] R. Zheng, J. Li, H. Chen, and Z. Huang, "A framework for authorship identification of online messages: Writing-style features and classification techniques," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 57, pp. 378–393, 2006.

[40] M. Koppel, J. Schler, and S. Argamon, "Computational methods in authorship attribution," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 60, pp. 9–26, 2009.

[41] H. V. Halteren, "Author verification by linguistic profiling: An exploration of the parameter space," *ACM Trans. Speech Lang. Process.*, vol. 4, pp. 1–17, 2007.

[42] S. Brin, J. Davis, and H. Garcia-Molina, "Copy detection mechanisms for digital documents," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, New York, 1995, pp. 398–409.

[43] N. Shivakumar and H. Garcia-Molina, "SCAM: A copy detection mechanism for digital documents," in *D-Lib Mag.*, 1995.

[44] K. J. Ottenstein, "An algorithmic approach to the detection and prevention of plagiarism," *SIGCSE Bull.*, vol. 8, no. 4, pp. 30–41, 1977.

[45] L. D. John, L. Ann-Marie, and H. S. Paula, "A plagiarism detection system," *SIGCSE Bull.*, vol. 13, no. 1, pp. 21–25, 1981.

[46] G. Sam, "A tool that detects plagiarism in Pascal programs," *SIGCSE Bull.*, vol. 13, no. 1, pp. 15–20, 1981.

[47] K. S. Marguerite, B. E. William, J. F. James, H. Cindy, and J. W. Leslie, "Program plagiarism revisited: Current issues and approaches," *SIGCSE Bull.*, vol. 20, pp. 224–224, 1988.

[48] Z. Ceska, "The future of copy detection techniques," in *Proc. YRCAS*, Pilsen, Czech Republic, pp. 5–10.

[49] D. I. Holmes, "The evolution of stylometry in humanities scholarship," *Lit Linguist Comput.*, vol. 13, pp. 111–117, 1998.

[50] O. deVel, A. Anderson, M. Corney, and G. Mohay, "Mining e-mail content for author identification forensics," *SIGMOD Rec.*, vol. 30, pp. 55–64, 2001.

[51] F. J. Tweedie and R. H. Baayen, "How variable may a constant be? Measures of lexical richness in perspective," *Comput. Humanities*, vol. 32, pp. 323–352, 1998.

[52] P. Clough, "Plagiarism in natural and programming languages: An overview of current tools and technologies," Dept. Comput. Sci., Univ. Sheffield, U.K., Tech. Rep. CS-00-05, 2000.

[53] P. Clough, (2003) Old and new challenges in automatic plagiarism detection. *National UK Plagiarism Advisory Service*. [Online]. Available: http://ir.shef.ac.uk/cloughie/papers/pas_plagiarism.pdf

[54] H. Maurer, F. Kappe, and B. Zaka, "Plagiarism—A survey," *J. Univ. Comput. Sci.*, vol. 12, pp. 1050–1084, 2006.

[55] L. Romans, G. Vita, and G. Janis, "Computer-based plagiarism detection methods and tools: An overview," presented at the Int. Conf. Comput. Syst. Technol., Rousse, Bulgaria, 2007.

[56] S. Argamon, A. Marin, and S. S. Stein, "Style mining of electronic messages for multiple authorship discrimination: First results," presented at the 9th ACM SIGKDD Int. Conf. Know. Discovery Data Mining, Washington, DC, 2003.

[57] M. Koppel and J. Schler, "Exploiting stylistic idiosyncrasies for authorship attribution," in *Proc. IJCAI Workshop on Computat. Approaches Style Anal. Synth.*, Acapulco, Mexico, 2003, pp. 69–72.

[58] S. Alzahrani, "Plagiarism auto-detection in arabic scripts using statement-based fingerprints matching and fuzzy-set information

retrieval approaches," M.Sc. thesis, Univ. Technol. Malaysia, Johor Bahru, 2008.
[59] C. Sanderson and S. Guenter, "On authorship attribution via Markov chains and sequence kernels," in *Proc. 18th Int. Conf. Pattern Recog.*, Hong Kong, 2006, pp. 437–440.
[60] P. Clough and M. Stevenson, "Developing a corpus of plagiarised short answers," in *Language Resources & Evaluation*, 2010.
[61] A. J. A. Muftah, "Document plagiarism detection algorithm using semantic networks," M.Sc. thesis, Faculty Comput. Sci. Inf. Syst., Univ. Teechnol. Malaysia, Johor Bahru, 2009.
[62] M. Roig, *Avoiding Plagiarism, Self-Plagiarism, and Other Questionable Writing Practices: A Guide to Ethical Writing*. New York: St. Johns Univ. Press, 2006.
[63] J. Bloch, "Academic writing and plagiarism: A linguistic analysis," *English for Specific Purposes*, vol. 28, pp. 282–285, 2009.
[64] I. Anderson, "Avoiding plagiarism in academic writing," *Nurs. Standard*, vol. 23, no. 18, pp. 35–37, 2009.
[65] K. R. Rao, "Plagiarism, a scourge," *Current Sci.*, vol. 94, pp. 581–586, 2008.
[66] E. Stamatatos, "Intrinsic plagiarism detection using character n-gram profiles," in *Proc. SEPLN*, Donostia, Spain, pp. 38–46.
[67] J. Karlgren and G. Eriksson, "Authors, genre, and linguistic convention," presented at the SIGIR Forum PAN, Amsterdam, The Netherlands, to be published.
[68] H. Baayen, H. van Halteren, and F. Tweedie, "Outside the cave of shadows: using syntactic annotation to enhance authorship attribution," *Lit. Linguist. Comput.*, vol. 11, pp. 121–132, Sep. 1996.
[69] M. Gamon, "Linguistic correlates of style: Authorship classification with deep linguistic analysis features," presented at the 20th Int. Conf. Comput. Linguist., Geneva, Switzerland, 2004.
[70] P. M. McCarthy, G. A. Lewis, D. F. Dufty, and D. S. McNamara, "Analyzing writing styles with Coh-Metrix," in *Proc. Florida Artif. Intell. Res. Soc. Int. Conf.*, Melbourne, FL, 2006, pp. 764–769.
[71] M. Jones, "Back-translation: The latest form of plagiarism," presented at the 4th Asia Pacific Conf. Edu Integr., Wollongong, Australia, 2009.
[72] S. Argamon, C. Whitelaw, P. Chase, S. R. Hota, N. Garg, and S. Levitan, "Stylistic text classification using functional lexical features: Research articles," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 58, pp. 802–822, 2007.
[73] L. Stenflo, "Intelligent plagiarists are the most dangerous," *Nature*, vol. 427, p. 777, 2004.
[74] M. Bouville, "Plagiarism: Words and ideas," *Sci. Eng. Ethics*, vol. 14, pp. 311–322, 2008.
[75] G. Tambouratzis, S. Markantonatou, N. Hairetakis, M. Vassiliou, G. Carayannis, and D. Tambouratzis, "Discriminating the registers and styles in the modern greek language—Part 1: Diglossia in stylistic analysis," Lit. Linguist. Comput., vol. 19, no. 2, pp. 197–220, 2004.
[76] G. Tambouratzis, S. Markantonatou, N. Hairetakis, M. Vassiliou, G. Carayannis, and D. Tambouratzis, "Discriminating the registers and styles in the modern Greek language—Part 2: Extending the feature vector to optimise author discrimination," *Lit. Linguist. Comput.*, vol. 19, pp. 221–242, 2004.
[77] C. K. Ryu, H. J. Kim, S. H. Ji, G. Woo, and H. G. Cho, "Detecting and tracing plagiarized documents by reconstruction plagiarism-evolution tree," in *Proc. 8th Int. Conf. Comput. Inf. Technol.*, Sydney, N.S.W., 2008, pp. 119–124.
[78] Y. Palkovskii, ""Counter plagiarism detection software" and "Counter counter plagiarism detection" methods," in *Proc. SEPLN*, Donostia, Spain, pp. 67–68.
[79] J. A. Malcolm and P. C. R. Lane, "Tackling the PAN'09 external plagiarism detection corpus with a desktop plagiarism detector," in *Proc. SEPLN*, Donostia, Spain, pp. 29–33.
[80] R. Lackes, J. Bartels, E. Berndt, and E. Frank, "A word-frequency based method for detecting plagiarism in documents," in *Proc. Int. Conf. Inf. Reuse Integr.*, Las Vegas, NV, 2009, pp. 163–166.
[81] S. Butakov and V. Shcherbinin, "On the number of search queries required for Internet plagiarism detection," in *Proc. 9th IEEE Int. Conf. Adv. Learn. Technol.*, Riga, Latvia, 2009, pp. 482–483.
[82] S. Butakov and V. Scherbinin, "The toolbox for local and global plagiarism detection," *Comput. Educ.*, vol. 52, pp. 781–788, 2009.
[83] A. Barrón-Cedeño, P. Rosso, and J.-M. Benedí, "Reducing the plagiarism detection search space on the basis of the kullback-leibler distance," in *Computational Linguistics and Intelligent Text Processing*, 2009, pp. 523–534.
[84] E. V. Balaguer, "Putting ourselves in SME's shoes: Automatic detection of plagiarism by the WCopyFind tool," in *Proc. SEPLN*, Donostia, Spain, pp. 34–35.
[85] T. Wang, X. Z. Fan, and J. Liu, "Plagiarism detection in Chinese based on chunk and paragraph weight," in *Proc. 7th Int. Conf. Mach. Learn. Cybern.*, Kunming, Beijing, China, 2008, pp. 2574–2579.
[86] A. Sediyono, K. Ruhana, and K. Mahamud, "Algorithm of the longest commonly consecutive word for plagiarism detection in text based document," in *Proc. 3rd Int. Conf. Dig. Inf. Manage.*, London, U.K., 2008, pp. 253–259.
[87] R. Řehurek, "Plagiarism detection through vector space models applied to a digital library," in *Proc. RASLAN*, Karlova Studánka, Czech Republic, pp. 75–83.
[88] Z. Ceska, "Plagiarism detection based on singular value decomposition," in Lecture Notes in Computer Science, vol. 5221, *Lecture Notes in Artificial Intelligence*, pp. 108–119, 2008.
[89] C. H. Leung and Y. Y. Chan, "A natural language processing approach to automatic plagiarism detection," in *Proc. ACM Inf. Technol. Educ. Conf.*, New York, 2007, pp. 213–218.
[90] J.-P. Bao, J.-Y. Shen, X.-D. Liu, H.-Y. Liu, and X.-D. Zhang, "Finding plagiarism based on common semantic sequence model," in *Advances in Web-Age Information Management*, 2004, pp. 640–645.
[91] M. Mozgovoy, S. Karakovskiy, and V. Klyuev, "Fast and reliable plagiarism detection system," presented at the Frontiers Educ. Conf., Milwaukee, WI, 2007.
[92] Y. T. Liu, H. R. Zhang, T. W. Chen, and W. G. Teng, "Extending Web search for online plagiarism detection," in *Proc. IEEE Int. Conf. Inf. Reuse Integr.*, Las Vegas, IL, 2007, pp. 164–169.
[93] D. Sorokina, J. Gehrke, S. Warner, and P. Ginsparg, "Plagiarism Detection in arXiv," in *Proc. 6th Int. Conf. Data Mining*, Washington, DC, 2006, pp. 1070–1075.
[94] N. Sebastian and P. W. Thomas, "SNITCH: A software tool for detecting cut and paste plagiarism," in *Proc. 37th SIGCSE Symp. Comput. Sci. Educ.*, New York, 2006, pp. 51–55.
[95] Z. Manuel, F. Marco, M. Massimo, and P. Alessandro, "Plagiarism detection through multilevel text comparison," in *Proc. 2nd Int. Conf. Automat. Prod. Cross Media Content for Multi-Channel Distrib.*, 2006, Washington, DC, pp. 181–185.
[96] W. Kienreich, M. Granitzer, V. Sabol, and W. Klieber, "Plagiarism detection in large sets of press agency news articles," in *Proc. 17th Int. Conf. Database Expert Syst. Appl.*, 2006, Washington, DC, pp. 181–188.
[97] N. Kang, A. Gelbukh, and S. Han, "PPChecker: Plagiarism pattern checker in document copy detection," in *Text, Speech and Dialogue*, 2006, pp. 661–667.
[98] J. P. Bao, J. Y. Shen, H. Y. Liu, and X. D. Liu, "A fast document copy detection model," in *Soft Computing - A Fusion of Foundations, Methodologies & Applications*, 2006, vol. 10, pp. 41–46.
[99] J. Bao, C. Lyon, and P. Lane, "Copy detection in Chinese documents using Ferret," in *Language Resources & Evaluation*, 2006, vol. 40, pp. 357–365.
[100] M. Mozgovoy, K. Fredriksson, D. White, M. Joy, and E. Sutinen, "Fast plagiarism detection system," in *Language Resources & Evaluation*, 2005, pp. 267–270.
[101] S. M. Alzahrani and N. Salim, "Plagiarism detection in Arabic scripts using fuzzy information retrieval," presented at the Student Conf. Res. Develop., Johor Bahru, Malaysia, 2008.
[102] S. M. Alzahrani, N. Salim, and M. M. Alsofyani, "Work in progress: Developing Arabic plagiarism detection tool for e-learning systems," in *Proc. Int. Assoc. Comput. Sci. Inf. Technol., Spring Conf.*, Singapore, 2009, pp. 105–109.
[103] E. Stamatatos, "Author identification: Using text sampling to handle the class imbalance problem," *Inf. Process. Manage.*, vol. 44, pp. 790–799, 2008.
[104] M. Koppel, J. Schler, and S. Argamon, "Authorship attribution in the wild," in *Language Resources & Evaluation*, 2010.
[105] L. Seaward and S. Matwin, "Intrinsic plagiarism detection using complexity analysis," in *Proc. SEPLN*, Donostia, Spain, pp. 56–61.
[106] L. P. Dinu and M. Popescu, "Ordinal measures in authorship identification," in *Proc. SEPLN*, Donostia, Spain, pp. 62–66.
[107] S. Benno, K. Moshe, and S. Efstathios, "Plagiarism analysis, authorship identification, and near-duplicate detection," in *Proc. ACM SIGIR Forum PAN'07*, New York, pp. 68–71.
[108] C. H. Lee, C. H. Wu, and H. C. Yang, "A platform framework for cross-lingual text relatedness evaluation and plagiarism detection," presented

at the 3rd Int. Conf. Innov. Comput. Inf. Control, Dalian, Liaoning, China, 2008.
[109] M. Potthast, A. Eiselt, B. Stein, A. BarrónCedeño, and P. Rosso. (2009). PAN Plagiarism Corpus (PAN-PC-09). [Online]. Available: http://www.uni-weimar.de/cms/medien/webis/ 2012.
[110] M. Potthast, A. Eiselt, B. Stein, A. BarrónCedeño, and P. Rosso. (2010, Sept. 18). PAN Plagiarism Corpus (PAN-PC-10). [Online]. Available: http://www.uni-weimar.de/cms/medien/webis/
[111] C. J. v. Rijsbergen, *Information Retrieval*. London, U.K.: Butterworths, 1979.
[112] G. Canfora and L. Cerulo, "A taxonomy of information retrieval models and tools," *J. Comput. Inf. Technol*., vol. 12, pp. 175–194, 2004.
[113] S. Gerard and J. M. Michael, *Introduction to Modern Information Retrieval*. New York: McGraw-Hill, 1986.
[114] C. D. Manning, P. Raghavan, and H. Schütze, "Boolean retrieval," in *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008, pp. 1–18.
[115] C. D. Manning, P. Raghavan, and H. Schütze, "Web search basics: Near-duplicates and shingling," in *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008, pp. 437–442.
[116] N. Heintze, "Scalable document fingerprinting," in *Proc. 2nd USENIX Workshop Electron. Commerce*, 1996, pp. 191–200.
[117] B. Stein, "Principles of hash-based text retrieval," in *Proc. 30th Annu. Int. ACM SIGIR*, Amsterdam, The Netherlands, 2007, pp. 527–534.
[118] S. Schleimer, D. Wilkerson, and A. Aiken, "Winnowing: Local algorithms for document fingerprinting," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, New York, 2003, pp. 76–85.
[119] C. D. Manning, P. Raghavan, and H. Schütze, "Scoring, term weighting and the vector space model," in *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2009, pp. 109–133.
[120] C. D. Manning, P. Raghavan, and H. Schütze, "Matrix decompositions and latent semantic indexing," in *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2009, pp. 403–417.
[121] H. Q. D. Chris, "A similarity-based probability model for latent semantic indexing," in *Proc. 22nd Annu. Int. ACM SIGIR*, Berkeley, CA, 1999, pp. 58–65.
[122] H. Chen, K. J. Lynch, K. Basu, and T. D. Ng, "Generating, integrating, and activating thesauri for concept-based document retrieval," *IEEE Expert, Intelli. Syst. Appl*., vol. 8, no. 2, pp. 25–34, Apr. 1993.
[123] Z. Ceska, "Automatic plagiarism detection based on latent semantic analysis," Ph.D. dissertation, Faculty Appl Sci., Univ. West Bohemia, Pilsen, Czech Republic, 2009.
[124] Y. Ogawa, T. Morita, and K. Kobayashi, "A fuzzy document retrieval system using the keyword connection matrix and a learning method," *Fuzzy Sets Syst*., vol. 39, pp. 163–179, 1991.
[125] V. Cross, "Fuzzy information retrieval," *J. Intell. Syst*., vol. 3, pp. 29–56, 1994.
[126] S. Zadrozny and K. Nowacka, "Fuzzy information retrieval model revisited," *Fuzzy Sets Syst*., vol. 160, pp. 2173–2191, 2009.
[127] D. Dubois and H. Prade, "An introduction to fuzzy systems," *Clinica Chimica Acta*, vol. 270, pp. 3–29, 1998.
[128] C. D. Manning, P. Raghavan, and H. Schütze, "Language models for information retrieval," in *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2009, pp. 237–252.
[129] C. D. Manning, P. Raghavan, and H. Schütze, "Probabilistic information retrieval," in *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2009, pp. 220–235.
[130] C. D. Manning, P. Raghavan, and H. Schütze, "Flat clustering," in *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2009, pp. 350–374.
[131] C. D. Manning, P. Raghavan, and H. Schütze, "Hierarchical clustering," in *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2009, pp. 378–401.
[132] S. K. Bhatia and J. S. Deogun, "Conceptual clustering in information retrieval," *IEEE Trans. Systems Man Cybern. B, Cybern*., vol. 28, no. 3, pp. 427–436, Jun. 1998.
[133] T. Kohonen, *Self-Organization and Associative Memory*, 3rd ed. Berlin, Germany: Springer-Verlag, 1982.
[134] M. K. M. Rahman, W. Pi Yang, T. W. S. Chow, and S. Wu, "A flexible multi-layer self-organizing map for generic processing of tree-structured data," *Pattern Recog*., vol. 40, pp. 1406–1424, 2007.
[135] S. K. Pal, S. Mitra, and P. Mitra, "Soft computing pattern recognition, data mining and web intelligence," in *Intelligent Technologies for Information Analysis*, N. Zhong and J. Lie, Eds. Berlin, Germany: Springer-Verlag, 2004.
[136] S. Kaski, T. Honkela, K. Lagus, and T. Kohonen, "WEBSOM: Self-organizing maps of document collections," *Neurocomput*., vol. 21, pp. 101–117, 1998.
[137] N. Ampazis and S. Perantonis, "LSISOM: A latent semantic indexing approach to self-organizing maps of document collections," *Neural Process. Lett*., vol. 19, pp. 157–173, 2004.
[138] S. Antonio, L. Hong Va, and W. H. L. Rynson, "CHECK: A document plagiarism detection system," in *Proc. ACM Symp. Appl. Comput*., San Jose, CA, 1997, pp. 70–77.
[139] G. Navarro, "A guided tour to approximate string matching," *ACM Comput. Surveys*, vol. 33, pp. 31–88, 2001.
[140] W. Cohen, P. Ravikumar, and S. Fienberg, "A comparison of string distance metrics for name-matching tasks," in *Proc. IJCAI Workshop Inf. Integr. Web*, Acapulco, Mexico, pp. 73–78.
[141] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, pp. 39–41, 1995.
[142] M. K. M. Rahman and T. W. S. Chow, "Content-based hierarchical document organization using multi-layer hybrid network and tree-structured features," *Expert Syst. Appl*., vol. 37, pp. 2874–2881, 2010.

Salha M. Alzahrani received the Bachelor's degree with first-class honors in computer science from the Department of Computer Science, Taif University, Taif, Saudi Arabia, in 2004 and the Master's degree in computer science from the University of Technology Malaysia (UTM), Johor Bahru, Malaysia, in 2009, where she is currently working toward the Ph.D. degree.

Her research interests include plagiarism detection, text retrieval, arabic natural language processing, soft information retrieval, soft computing applied to text analysis, and data mining.

Ms. Alzahrani received the Vice Chancellor's award for excellent academic achievement from UTM in 2009. Since 2010, she has been the Coordinator of the web activities of the IEEE Systems, Man, and Cybernetics Technical Committee on Soft Computing.

Naomie Salim received the Bachelor's degree in computer science from the University of Technology Malaysia (UTM), Johor Bahru, Malaysia, in 1989, the Master's degree from the University of Illinois, Urbana-Champaign, in 1992, and the Ph.D. degree from the University of Sheffield, Sheffield, U.K., in 2002.

She is a currently a Professor and the Deputy Dean of Postgraduate Studies with the Faculty of Computer Science and Information System, UTM. Her research interests include information retrieval, distributed databases, plagiarism detection, text summarization, and chemoinformatics.

Ajith Abraham (M'96–SM'07) received the Ph.D. degree from Monash University, Melbourne, Australia, in 2001.

He has worldwide academic experience of over 12 years with formal appointments with different universities. He is currently a Research Professor with the VSB-Technical University of Ostrava, Ostrava-Poruba, Czech Republic, and is the Director of the Machine Intelligence Research Labs, Seattle, WA. He has authored or coauthored more than 700 research publications in peer-reviewed reputed journals, book chapters, and conference proceedings. His research interests include machine intelligence, network security, sensor networks, e-commerce, Web intelligence, Web services, computational grids, and data mining applied to various real-world problems.

Dr. Abraham has been the recipient of several Best Paper Awards and has also received several citations. He has delivered more than 50 plenary lectures and conference tutorials. He is the Co-Chair of the IEEE Systems Man and Cybernetics Society Technical Committee on Soft Computing. He has been on the Editorial Boards of more than 50 international journals and was a Guest Editor of more than 40 special issues on various topics. He is an active Volunteer for the ACM/IEEE. He has initiated and is also involved in the organization of several annual conferences sponsored by the IEEE/ACM: HIS (11 years), ISDA (11 years), IAS (seven years) NWeSP (seven years), NaBIC (three years), SoCPaR (three years), CASoN (three years), etc., besides other symposiums and workshops. He is a Senior Member of the IEEE Systems Man and Cybernetics Society, ACM, the IEEE Computer Society, IET (U.K.), etc.