Plagiarism Detection Algorithm Model Based on NLP Technology

Ahmed A. Elngar[1*], Mohamed Gamal[2], Amar Fathy[3], Basma Moustafa[4], Omar Mahmoud[5] and Mohamed Shaban[6]

[1] Faculty of Computers and Artificial Intelligence, Beni-Suef University, Beni Suef, City, 62511, Egypt, Founder of Scientific Innovation Research Group (SIRG), , E-mail: elngar_7@yahoo.co.uk

[2] Department of Computer Science, Scientific Innovation Research Group (SIRG) member, Beni Suef University of Computers and Artificial Intelligence, E-mail: peseesabim@gmail.com

[3] Department of Computer Science, Scientific Innovation Research Group (SIRG) member, Beni Suef University of Computers and Artificial Intelligence, E-mail: fathyamar78@gmail.com

[4] Department of Computer Science, Scientific Innovation Research Group (SIRG) member, Beni Suef University of Computers and Artificial Intelligence, E-mail: basmamoustafa31@gmail.com

[5] Department of Computer Science, Scientific Innovation Research Group (SIRG) member, Beni Suef University of Computers and Artificial Intelligence, E-mail: mora50017@gmail.com

[6] Department of Computer Science, Scientific Innovation Research Group (SIRG) member, Beni Suef University of Computers and Artificial Intelligence, E-mail: muahmmedshaban153@gmail.com

[*]**Corresponding Author: Omar Mahmoud,** E-mail: mora50017@gmail.com

Abstract: We can bear in mind that each of us has plagiarized a text without realizing that it was plagiarism, Plagiarism can happen in Articles, Papers, Researches, literature, music, software, scientific, newspapers, websites, Master and PHD Thesis and many other fields, So plagiarism has become serious major problem to teachers, researchers and publishers, There are divergent opinions about how to define plagiarism and what makes plagiarism serious.

So, the detecting plagiarism is very important, so in this survey we explicate the concept of "plagiarism" and provide an overview of different plagiarism software and tools to solve the plagiarism problem, and will discuss the plagiarism process, types and detection methodologies. We can define that plagiarism is the brief and the description of this sentence "someone used someone else's mental product (such as its texts, ideas, or privacy). We suggest that what makes plagiarism so reprehensible is that it distorts scientific credit. In addition, intentional plagiarism indicates dishonesty. Moreover, there are a number of possible negative consequences of plagiarism. So we just create a framework for external plagiarism detection in which a some NLP processes are applied to process a set of suspicious and original documents, we have classified the different plagiarism detection techniques based on Lexical, Semantic, Syntactic and grammar analysis algorithms, And all of these algorithms precedes it NLP processing.

Keywords: *plagiarism, NLP, detection methodologies, Lexical Analysis, Semantic Analysis, NLTK, LSA, PLSA, LDA.*

Received: June 23, 2020 Revised: August 22, 2020 Accepted: October 16, 2020

DOI: 10.5281/zenodo.4099751

1. Introduction

As a result of daily massive use of the internet in our life, it has led to crimes spread. Plagiarism has become an easy way for people to plagiarize other's text, property and violate their rights. Plagiarism is a known and growing issue in the academic world. For some magazines, it's really a serious problem, as it up to a third of the published papers that contain plagiarism (Zhang 2010; Butler 2010). First, we will explain the concept of "plagiarism", that is, we will provide an analysis of the concept aimed at further clarification. Through critical analysis, we come up with what we consider to be an improved definition. We can define plagiarism as "when someone uses someone else's ideas, words or work and pretends to be their ideas." And in scientific definitions, "plagiarism is the appropriation of another person's ideas, operations, results, or words without giving appropriate credit" (US Federal Policy on Research Behavior). So it seems that the basic ideas are that someone deliberately takes someone else's work, whether in the form of an idea, or text, or data, or results, or method, and displays them as its own rather than granting credit to the person who used his ideas or results, or words. There are several kinds of plagiarism detection methods.

Plagiarism is divided into two types which are Intrinsic and Extrinsic Plagiarism which intrinsic plagiarism detection consists in finding plagiarized passages within a document without access to potential original texts. And there Four kinds for intrinsic plagiarism that considered in the detection of plagiarism, First Near Copies: is a form of plagiarism that the source without citing it in the right way, second Disguised: is a form of plagiarism that cites a part or whole document, restructures the sentences and changes the words with the similar words, third Translated: is a form of plagiarism that translates source document into foreign language, fourth Idea: is a form of plagiarism that changes everything using its own structure and words but discuss exactly the same topic with the source ones. We can say the best description of a plagiarism is either a 'copy and paste' for a text even if the source was cited or a change in some words by taking the meaning without citing the source and all of that falls under Semantic analysis and this is hard types of Algorithms in plagiarism detection and most complex one.

The main topic of this Survey is to find the best way to detect semantic analysis in plagiarism which occurs on the meaning and making use of synonyms and replace it instead of the original words. And at the beginning of that process, we must prepare the corpus for detection stage This survey aims also to apply a NLP pre-processing for corpus by using segmentation, tokenization, lowercase, stop word, stemming and POS tags, then tested whether the research enters under the specialization of computer science or not

Received: June 23, 2020 Revised: August 22, 2020 Accepted: October 16, 2020
DOI: 10.5281/zenodo.4099751

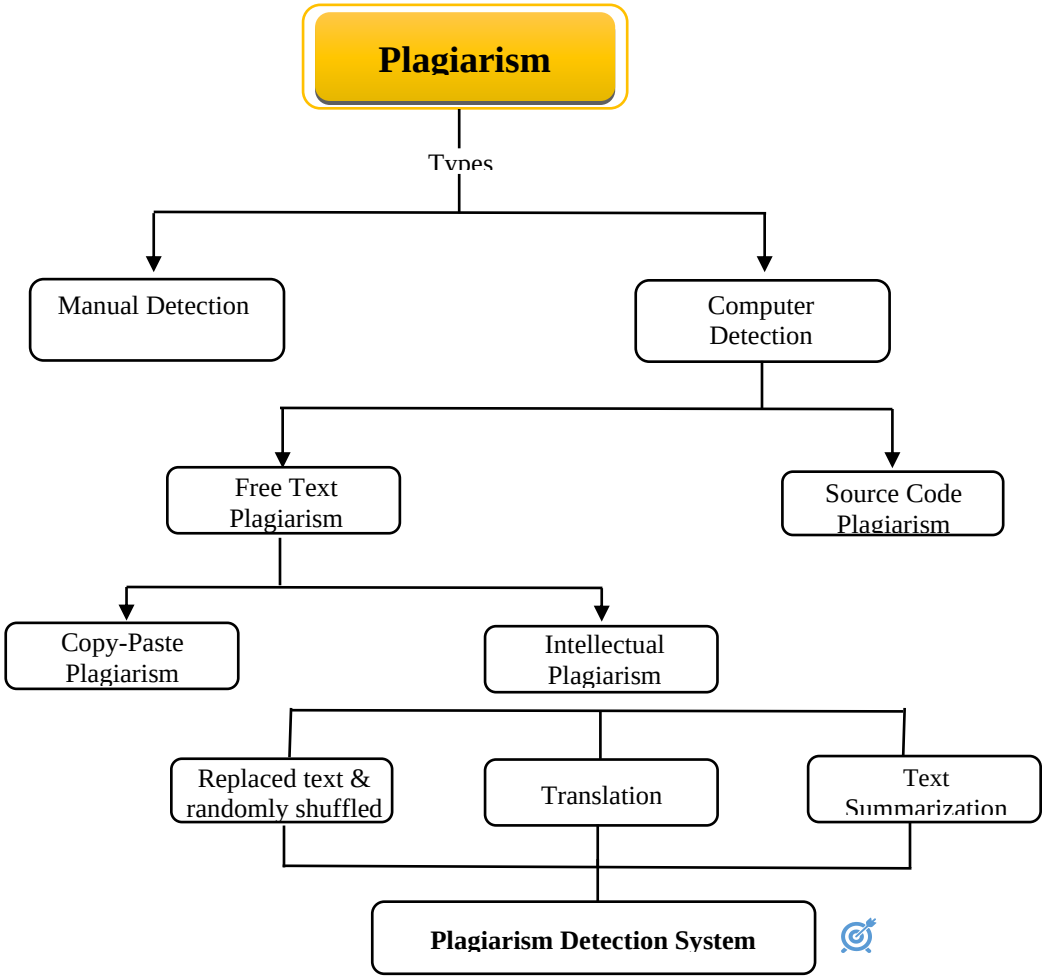


Fig 1: General Classification to Plagiarism Detection

2. Implementation Methodology

We now turn to explaining some methods of text detection implementation on whether that text is considered plagiarism or just unique text, in general the plagiarism system consists of four main approaches or phases, each phase consists of some steps.

- Pre-processing
- Lexical Analysis Algorithm
- Semantic Analysis Algorithm
- Grammar Analysis Algorithm

First of all, and before anything, we should make sure that our input "corpus" is text and not something else.

Received: June 23, 2020 Revised: August 22, 2020 Accepted: October 16, 2020
DOI: 10.5281/zenodo.4099751

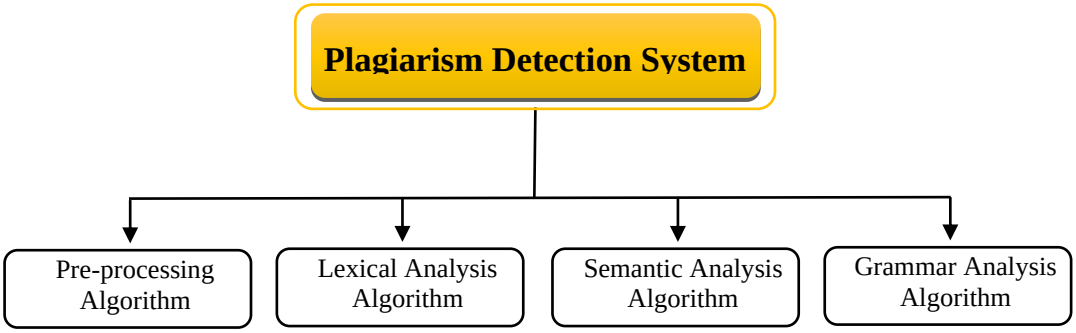


Figure 2. Approaches of Plagiarism Detection System

2.1 Pre-processing Stage

This stage consists of several steps as shown in Figure 3.

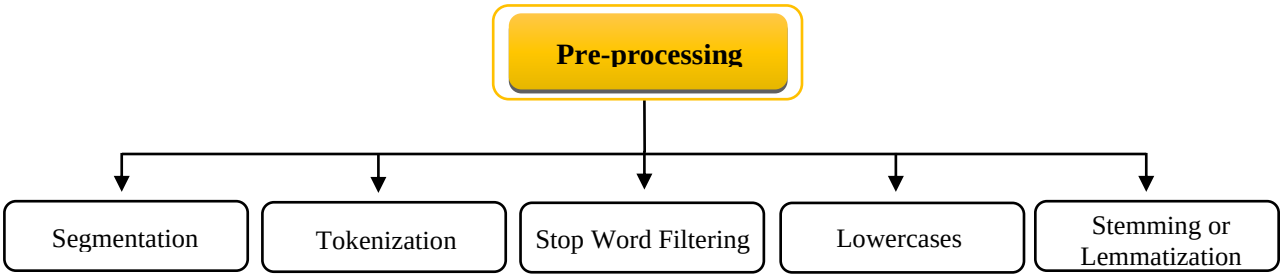


Figure 3. NLP Pre-processing stages in plagiarism System

We have a Python library that enables us to do the make the pre-processing on our framework, which makes it easier for us to process the pre-processing stages.

2.2 NLTK (Python Toolkit)

NLTK is considered that an open- source python library and NLTK is an abbreviation to "Natural Language Toolkit" to make NLP processes in python easier to build. There are many built in functions that are included inside this NLTK library, according to NLTK 3.5 documentation Natural Language Toolkit is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WorldNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum. [1]

NLTK has been called "a wonderful tool for teaching and working in, computational linguistics using Python," and "an amazing library to play with natural language."

2.2.1 Segmentation:

Segmentation or separation is considered the first stage of the NLP pre-processing, at this Stage the input text or corpus is divided into equal sentences, separating main parts from the corpus can be manually or programmatically, but in this survey, segmentation will be programmatically to prepare it for the next stage. Segmentation is the task of dividing a text into linguistically-meaningful units, Sentence words consisting of one or more characters, since the words and sentences identified at this stage are the fundamental units passed to further processing stages such as morphological analyzers, part-of-speech

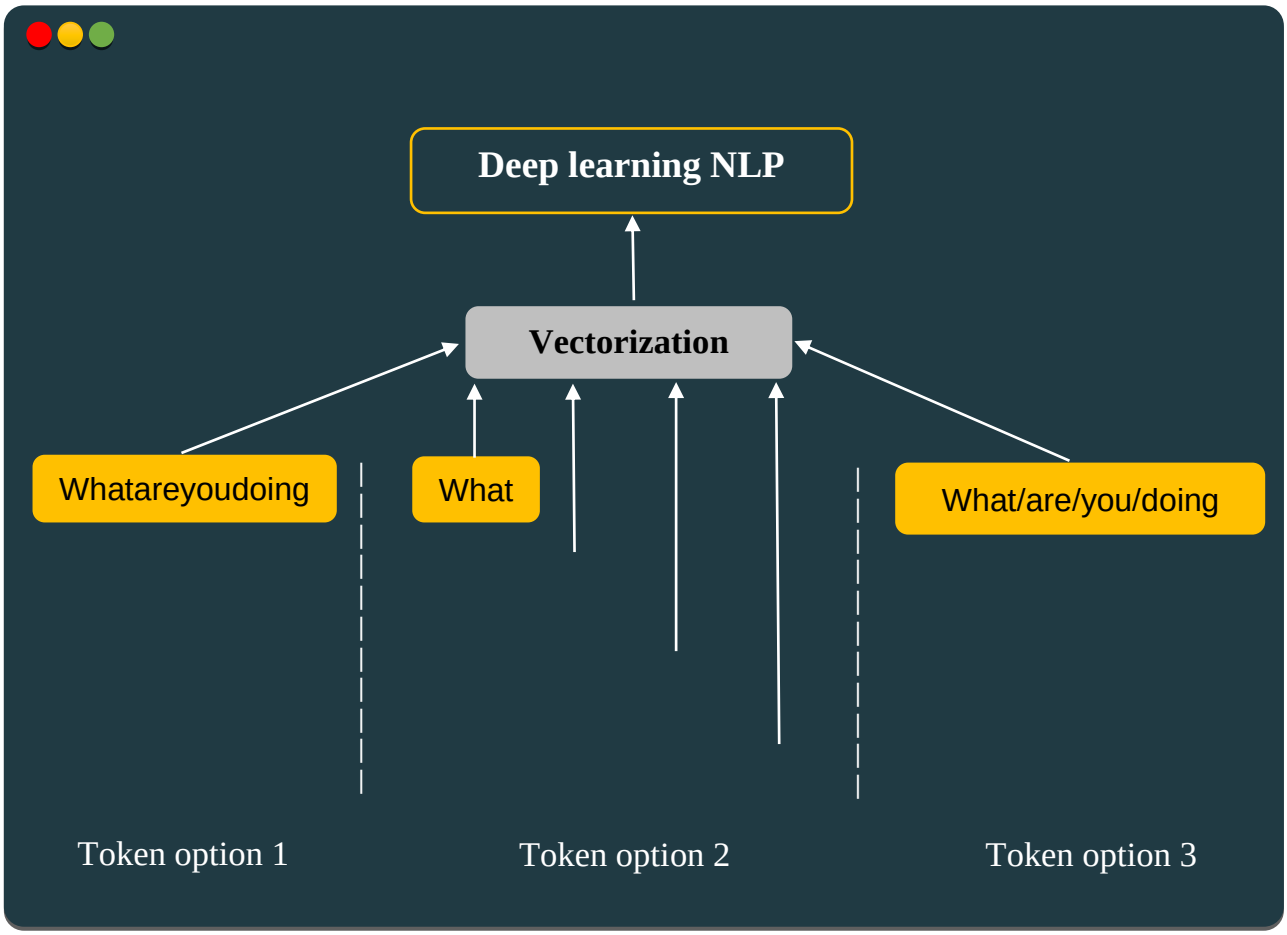
Received: June 23, 2020 Revised: August 22, 2020 Accepted: October 16, 2020
DOI: 10.5281/zenodo.4099751

taggers and parsers… According to Practically, sentence, and word separation or segmentation can't be performed successfully independent from one another. E.g. an essential subtask in both word and sentence segmentation for English is identifying abbreviations, because a period can be used in English to mark an abbreviation as well as to mark the end of a sentence. In the case of a period marking an abbreviation, the period is usually considered a part of the abbreviation token, whereas a period at the end of a sentence is usually considered a token in and of itself. Tokenizing abbreviations is complicated further when an abbreviation occurs at the end of a sentence, and the period marks both the abbreviation and the sentence boundary. [2]

2.2.2 Tokenization: (Tokenization && Punctuation Removal)

It is considered that the second and important stage of NLP pre-processing through which sentences is divided into set of tokens, because the input of this stage is a text of the document that consists of sentences, Each "entity" that is a part of whatever was split up based on rules. For examples, each word is a token when a sentence is "tokenized" into words. Each sentence can also be a token, if you tokenized the sentences out of a paragraph.

Etc. ['Hello Ahmed, how old are you?', 'The Football are a beautiful sport, and Java is awesome.', 'You are cute today.'] as we see some of sentences have symbols and delimiters, we should get rid of it at this stage (like. @, $, %, &, *, #, {,}, >, <, “, ‘,;, :), And this is called Punctuation Removal, in which punctuation is deleted from sentences to be input of the next stage.



Received: June 23, 2020 Revised: August 22, 2020 Accepted: October 16, 2020
DOI: 10.5281/zenodo.4099751

As we saw at the previous diagram there are more than a way that enable us to tokenize our corpus, the first option is not ideal since all the words are simply stacked together into one token and that is enough to be more difficult in processing, the second option separate the sentence into small tokens "word tokens", the third option uses one token but adds the "/" symbol to try and differentiate between words. These models do not have knowledge of language. So they cannot learn from text if they don't understand anything about the structure of language.

2.2.3 Stop Words Filtering or Removing:

The English language contains some words, which in turn constitute a meaning that has no significance for the plagiarism detection process, just its use is to prevent the following processing being over influenced by very frequent words, Words like pronouns, conjunctions and prepositions.

Delete functional words "pronouns, Adverbs, Adjectives, Article. (the, an, a, as, there, such as) The result of this stage is a text free of stop words, the usual way of determining what counts as a stop word is just to use a dictionary that lists them for Czech. Stop words will cost us a lot of space in our database and processing and also taking up valuable processing time, given that we will not need it to check plagiarism, so we store a list of words full of words we have mentioned above. Stop words are considered that the most commonly written word in the English language. The plurality of sentences therefore shares a similar percentage of stop words. Stop word existence and using it in processing and detection will not add any additional meaning.

If both are normal and lack any useful knowledge, why we do not delete it!?

With Stop Words	Without Stop Words
The Greed has poisoned men's souls, has barricaded the world with hate, has goose-stepped us into misery and bloodshed	Greed poisoned men's souls, barricaded hate, goose-stepped us misery bloodshed

2.2.4 Lowercases:

After completing stop word filtering stage, We will find some words with capital letters, to reduce the size of the vocabulary of our text results, we lowercase the text, we just have to replace all capital letters with small letters, Meaning that we puts all letters in lower case, we have to replace every upper cases with lowercases letters to generalize the matching and can identify with the text to facilitate the detection of plagiarism.

2.2.5 Stemming | Lemmatization:

Received: June 23, 2020 Revised: August 22, 2020 Accepted: October 16, 2020
DOI: 10.5281/zenodo.4099751

Stemming, which attempts to normalize words or sentences into its base form or root form, it's another preprocessing step that we can perform. Different types of words and sentences often having the same meaning in the English language. Stemming is important, not just for making detection broader and increasing retrieval, but also as a tool for compressing corpus size. Another way to reduce text or input size is simply to index fewer words. For search purposes, some words are more important than others. A significant reduction in index size can be achieved by stemming only the more important words.

Stemming is a way to account for these types, furthermore, it will help us shorten the sentences and shorten our lookup. For example, consider the following sentence:

- I will buy a new book.
- I have booked a ticket.

Rules	Examples
• s ⟶ • ss ⟶ ss • ies ⟶ i • sses ⟶ ss	• Dogs ⟶ Dog • Actress ⟶ Actress • Activities ⟶ Activity • Actresses ⟶ Actress

Porter Stemmer:

Porter's stemmer is the oldest stemmer for English language in which has five heuristic phases of word reductions applied sequentially, it has an algorithm removes suffixes from words.

Protect
Protected
Protecting
Protection
Protections

☆ **Porter Stemmer Algorithm**

We have in English directory a vowel letters such as (A, E, I, O and U) and other than Y preceded by a consonant. So a consonant will be denoted by c, a vowel by v. A list ccc… of length greater than 0 will be denoted by C, and a list vvv… of length greater than 0 will be denoted by V, we can see any word or part of a word, therefore has one of four forms

CVCV … C
CVCV … V

Received: June 23, 2020 Revised: August 22, 2020 Accepted: October 16, 2020
DOI: 10.5281/zenodo.4099751

VCVC … C
VCVC … V

Step 1c

(*v*) Y -> I	happy	-> happi
	sky	-> sky

Step 1 deals with plurals and past participles. The subsequent steps are much more straightforward.

Step 2

(m>0) ATIONAL	->	ATE	relational	->	relate
(m>0) TIONAL	->	TION	conditional	->	condition
			rational	->	rational
(m>0) ENCI	->	ENCE	valenci	->	valence
(m>0) ANCI	->	ANCE	hesitanci	->	hesitance
(m>0) IZER	->	IZE	digitizer	->	digitize
(m>0) ABLI	->	ABLE	conformabli	->	conformable
(m>0) ALLI	->	AL	radicalli	->	radical
(m>0) ENTLI	->	ENT	differentli	->	different
(m>0) ELI	->	E	vileli	- >	vile
(m>0) OUSLI	->	OUS	analogousli	->	analogous
(m>0) IZATION	->	IZE	vietnamization	->	vietnamize
(m>0) ATION	->	ATE	predication	->	predicate
(m>0) ATOR	->	ATE	operator	->	operate

Step 3

(m>0) ICATE	->	IC	triplicate	->	triplic
(m>0) ATIVE	->		formative	->	form
(m>0) ALIZE	->	AL	formalize	->	formal
(m>0) ICITI	->	IC	electriciti	->	electric
(m>0) ICAL	->	IC	electrical	->	electric
(m>0) FUL	->		hopeful	->	hope
(m>0) NESS	->		goodness	->	good

Step 4

(m>1) AL	->		revival	->	reviv
(m>1) ANCE	->		allowance	->	allow
(m>1) ENCE	->		inference	->	infer
(m>1) ER	->		airliner	->	airlin
(m>1) IC	->		gyroscopic	->	gyroscop

```
Step 5a

    (m>1) E     ->                  probate         ->  probat
                                    rate            ->  rate
    (m=1 and not *o) E ->           cease           ->  ceas

Step 5b

    (m > 1 and *d and *L) -> single letter
                                    controll        ->  control
                                    roll            ->  roll
```

2.2.5.1 Lemmatization:

Somehow similar to Stemming, as it maps several words into one common root, and the output of this stage is a proper word, Lemmatization makes use of the context to show word meaning. This is particularly important for languages that have rich systems of inflexion.

Etc. A lemmatize should map gone, going and went into go

3. Plagiarism Detection:

3.1. Lexical Analysis

Lexical analysis is the applied of NLP in plagiarism detection that parses sentences into tokens that will be compared to other tokens lexically. And in this algorithm stage will use Vectorization because python will not understand these characters, but it needs some number in order to work, which Vectorization is just the process of tokens clean words and then converting it into some Numerical representation or feature vector (vector of numerical feature that represent an object) or we can say that is the process of encoding texts as integers to create vector, It is considered an initial stage of pre-processing.

There are three types of Vectorization:

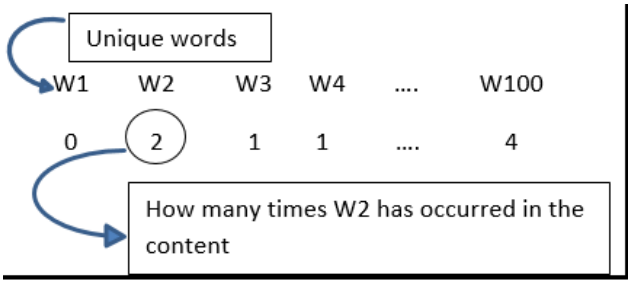
Received: June 23, 2020 Revised: August 22, 2020 Accepted: October 16, 2020
DOI: 10.5281/zenodo.4099751

- Count Vectorization
- N-gram
- TF-IDF

Which Count Vectorization is traditional technique which creates a document term matrix. For example, we have some sentences, and by pre-processing operation we can get tokens and then extract unique words from this tokens assume they are W1, W2, W3… W100

After that check each sentence and compare its words with the unique words, then we can get unique ID for each sentence. We can import it by:

from sklearn.feature_extraction.text import CountVectorizer



and N-gram is create a document-term matrix by making comparisons based on cutting the sentences into words with length n. position of the next n-gram will start from last shift's position n-gram and parameter n depends on the division that will be used by the n-gram method
There are three type of n-gram:

- bigram
- trigram
- 4-gram

Which bigram means n = 2, trigram means n = 3 and 4-gram means n = 4.
For example, we have sentence "I am studying NLP", by N-gram will be:

- Bigram: ("I am" , "am studying" , "studying NLP")
- Trigram: ("I am studying" , "am studying NLP")
- 4-gram: ("I am studying NLP")

And in the end we will use TF-IDF method to detect plagiarism, which TF-IDF (term frequency–inverse document frequency) is a numerical statistic that is intended to reflect how important a word is to a document in a corpus. The value of TF-IDF increases with the number of times a word appears in the document, we use TF-IDF in next algorithms, to calculate the weight we use this condition:

$$\boldsymbol{W_{i,j} = tf_{i,j} \times \log \frac{N}{df_i}}.$$

Which $tf_{i,j}$ is number of times a term I occur in a document j divided by total number of terms in j, for example if we have sentence which contain 10 words and a word occurred 3 times in this sentence then it will be 3/10, And N is Total number of documents in the corpus, And df_i is the number of documents which contained the term I.

Received: June 23, 2020 Revised: August 22, 2020 Accepted: October 16, 2020
DOI: 10.5281/zenodo.4099751

3.2 Plagiarism Detection Algorithm based on Semantic Analysis:

sematic analysis deal with the meaning of the words, "it parses sentence into terminal tokens (words) which might have another words with the same meaning in order to be able to detect plagiarism"[3]. There are more than one method using semantic analysis in plagiarism detection like Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), Latent Dirichlet Analysis (LDA) and Hyperspace Analog to Language (HAL).

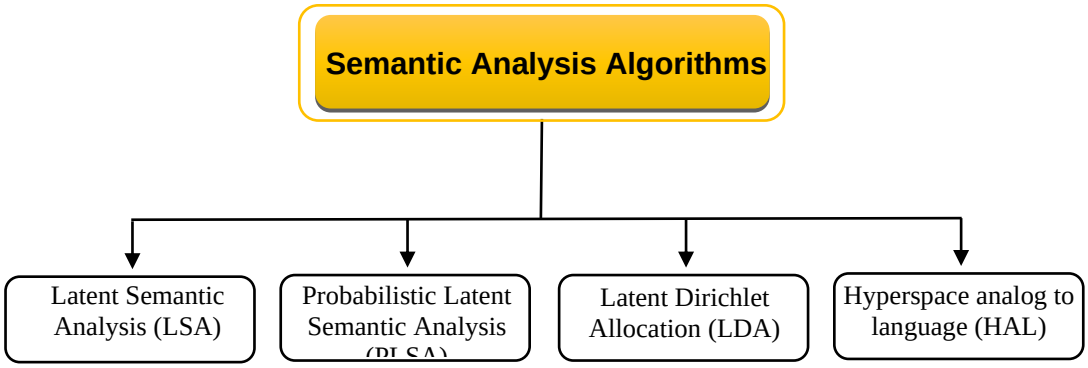


Figure 5. Classification and types of Semantic Analysis Algorithms

3.2.1 Latent Semantic Analysis (LSA):

LSA is a technique in natural language processing, also known as Latent Semantic Indexing (LSI), it is measuring similarity between two words by calculate the value of the cosine of two vectors, if value close to 1 represent similar words, but if value close to 0 represent dissimilar words. "Analyzing the content of the document, a paragraph, a sentence and so on" [4]. And then build a dimensional matrix rows represent unique words and columns represent document. Then used singular value decomposition (SVD) is a mathematical technique used to reduce the number of rows by divide the matrix into three matrix two orthogonal matrix and one diagonal matrix.

(SVD) ⊚

$N = U\sum V^T$

$\sum$ is square diagonal of size r × r

U is an m × r matrix

V is an n × r matrix

$\cos(vw1,vw2) = \sum_{n=1}^{\infty}(vw1i \cdot vw2i)/(|vw1|.|vw2|)$

3.2.2 Probabilistic Latent Semantic Analysis (PLSA):

PLSA Probabilistic Latent Semantic Analyses is a statistical technique, also known as Probabilistic Latent Semantic Indexing (PLSI), it is come from the standard Latent Semantic Analyses, but Latent Semantic Analysis (LSA) perform a Singular Value Decomposition and PLSA based on mixture decomposition derived from latent class model. PLSA is not a generative model because can't compute probability of a

Received: June 23, 2020 Revised: August 22, 2020 Accepted: October 16, 2020
DOI: 10.5281/zenodo.4099751

new document. "The starting point for Probabilistic Latent Semantic analysis is a statistical model, which also called aspect model" [5].

It is a latent variable Z which introduce and relate to document d. It based on the probabilistic and calculate the probability of each word. It use EM algorithm which avoid over fitting training data.

Aspect model →

$$p(w|d) = \sum_{z\in Z} p(w|c)p(z|c)$$

$$p(d|w) = \sum_{z\in Z} p(c)p(d|c)p(w|c)$$

- d is the document index variable.
- c is a word's topic drawn from the document's topic distribution, P (c | d).
- And w is a word drawn from the word distribution of this word's topic, P (w | c).
- The d and w are observable variables, the topic c is a latent variable.

EM algorithm →

$$p_\beta(c|d,w) = \frac{p(C)[p(d|C)p(w|C)]}{\sum_{z`} p(C`)[p(d|C`)p(w|C`)]}$$

3.2.3 Latent Dirichlet Allocation (LDA):

LDA is an algorithm that used to classify text in a document to a particular topic. Each document is modeled as a multinomial distribution of topics and each topic is modeled as a multinomial distribution of words. In LDA each document is considered to have more than one topic that are assigned to it via LDA. This is identical to PLSA, except that in LDA it is assumed that the topic distribution have a spare Dirichlet prior. In fact we make PLSA a generative model by imposing a Dirichlet prior on the model parameters. So LDA is Bayesian version of PLSA, and can achieve the same goal as PLSA for text mining purposes.

Steps of LDA:

- The number of words that appear in the document is determined.
- A topic mixture over a fixed set of topics is chosen for the document.
- A topic is chosen based on the multinomial distribution of the document.
- Now a word is chosen based on the multinomial distribution of the topic.

٣,٢,٤ Hyperspace analog to language (HAL):

HAL also known as semantic model or memory. The purpose of it is to capture the meaning of words by using co-occurrence of word in some text corpus. "In HAL the word is represented by a vector of other words, but in LSA the word is represented by vector of document" [6]. It is based on a sliding window. It represented as a matrix with L-size where a single word is represented by a row vector and a column vector

Received: June 23, 2020 Revised: August 22, 2020 Accepted: October 16, 2020

DOI: 10.5281/zenodo.4099751

that capture the information before and after the word. It is premised on context surrounding a word providing important information about its meaning" [7]. "A sliding L-size window moves word-by-word over a large text corpus" [8].

	advantages	disadvantages

$$HAL(t`|t) = \sum_{k=1}^{k} w(k)n(t, k, t`)$$

Where n (t, k, t0) is the number of times term t0 occurs a distance k away from t, and w(k)= K − k + 1 denotes the strength of relationship between the two terms given k [9].

Received: June 23, 2020 Revised: August 22, 2020 Accepted: October 16, 2020

DOI: 10.5281/zenodo.4099751

☆ ***Comparison of the advantages and disadvantages of semantic analysis algorithm***

Latent Semantic Analysis (LSA)	• Easy to implement, understand and use. • High performance • Runtime: it is faster compared to other dimensional reduction models • It is not sensitive to starting conditions (like neural network)	• It is a linear model. • Since it is a distributional model, so not an efficient representation. • Representation is dense.
Probabilistic Latent Semantic Analysis (PLSA)	• well defined probabilities • Better model selection and complexity control. • In order to avoid overfitting,	• One disadvantage of PLSA is that the EM algorithm like most iterative algorithms, converges only locally.
Latent Dirichlet Allocation (LDA)	• LDA is probabilistic model with interpretable topics. • Each document can have more than one topic.	• complexity of the model • Lack of observation, so more flexible model need to more observations to infer.
hyperspace analog to language (HAL)	• They use learning procedures that scale up to real word language.	• Have several limitation , including high processing cost ,

Received: June 23, 2020 Revised: August 22, 2020 Accepted: October 16, 2020

DOI: 10.5281/zenodo.4099751

3.3 Grammar Analysis Algorithms:

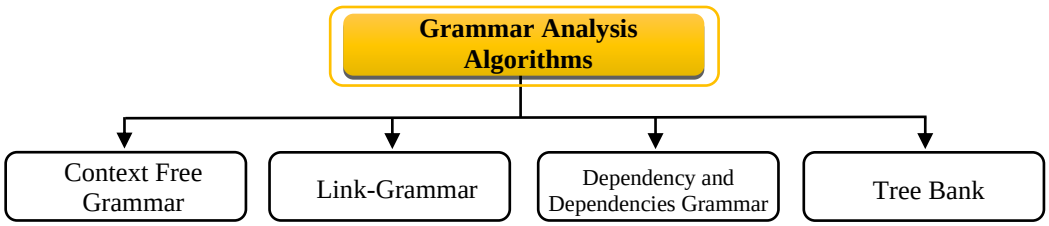


Figure 5. Classification and types of Grammar Analysis Algorithms

3.3.1 Context Free Grammar:

Context Free Grammar (CFG) is a series of recursive rules used to create a string patterns.it can be described all regular language and more, but they cannot describe all possible language. It can be studied in most fields of computer science, compiler design and linguistic.it can be used to describe programming language and parser programs [10].

Two parse trees that describe CFGs that generate the string "x + y * z". Source: Context-free grammar

Received: June 23, 2020 Revised: August 22, 2020 Accepted: October 16, 2020
DOI: 10.5281/zenodo.4099751

It can generate a context free language. they can do this by taking a series of variables which can defined recursively, in terms of each other, by a set of production rules it was called by this name, because any of production rules in the grammar can be regardless of context.it does not depend on any other symbol.

What are the Components of context free grammar?

- A set of terminal symbols which are the strings that appear in the language created by the grammar.it always appears in the right-hand side of the rule.
- A set of non-terminal symbol (variable): it is a place holder of terminal symbol that it can be created by terminal symbol.it always appears in the left-hand side of production rules.
- A set of production rules: it is replacing non-terminal symbol and can express the form of it → variable →character of variable and terminal.
- A beginning symbol→it is used special non-terminal symbol that exists in the starting of strings [11].

To create string from context free grammar:

1. Start the string from the begging symbol.
2. Implement one of the production rules to the begin symbol by replace it with the right-hand side.
3. Repeat all previous steps to replace all non-terminal with the right-hand side, until non-terminal replaced with terminal symbol.

Formal definition: -

Context free grammar R can be defined by four tuples as:

$$R= (V, T, P, S)$$

-R describes the grammar.
-T describes a finite set of -terminal symbols where v ∩T=null.
-V refer to a finite set of non-terminal symbols.
-P describes a set of production rules, P: $V\rightarrow (V \cup T)^*$.
-S is the start symbol. [12]

Exemple :-

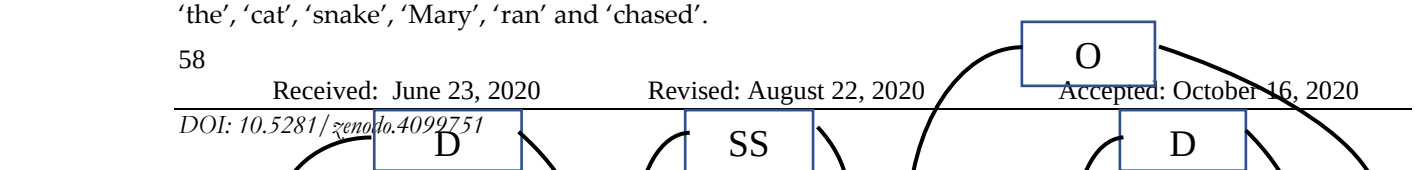
- The grammar ({X}, {x, y, z}, P, X), P : X → xX, X → xyz.
- The grammar ({S, x, y}, {x, y}, P, S), P: S → xSx, S → ySy, S → ε
- The grammar ({S, V}, {0, 1}, P, S), P: S → 00S | 11V, V → 00V |

3.3.2 Link Grammar:

Link Grammar is an official grammar system identified by Daniel D.K. Sleator Davy Timberly in 1991 based on the natural language property that provided for it if arcs are drawn connecting each pair of words related to each other, then the arches will not be curved. The authors have written grammar for nearly seven hundred Definitions embody many of the phenomenon of English grammar and have evolved. An algorithm for analyzing sentences according to the rules of the specified language.

Basic Definition and Notions: -

There are a set of words in link grammar like the terminal symbols of the grammar. A series of words is a sentence of the language identify by the grammar if there exists an approach to draw arcs among the words. Each word has a linking requirement. The connectivity (linking) requirements of all word are existed in a dictionary. This figure explains linking requirement of words_ with a simple dictionary for the words 'a', 'the', 'cat', 'snake', 'Mary', 'ran' and 'chased'.



Received: June 23, 2020 Revised: August 22, 2020 Accepted: October 16, 2020
DOI: 10.5281/zenodo.4099751

I think each words in connectors is coming out. Connectors either indicating the left or the right direction. A link between connectors will satisfy by connecting from left direction to the right direction. Plugging a pair of connectors together used to draw a link between the pair of words. The figure shows how the linking requirements are met (satisfied) in the sentence 'The Cat Chased a Snake' [16].

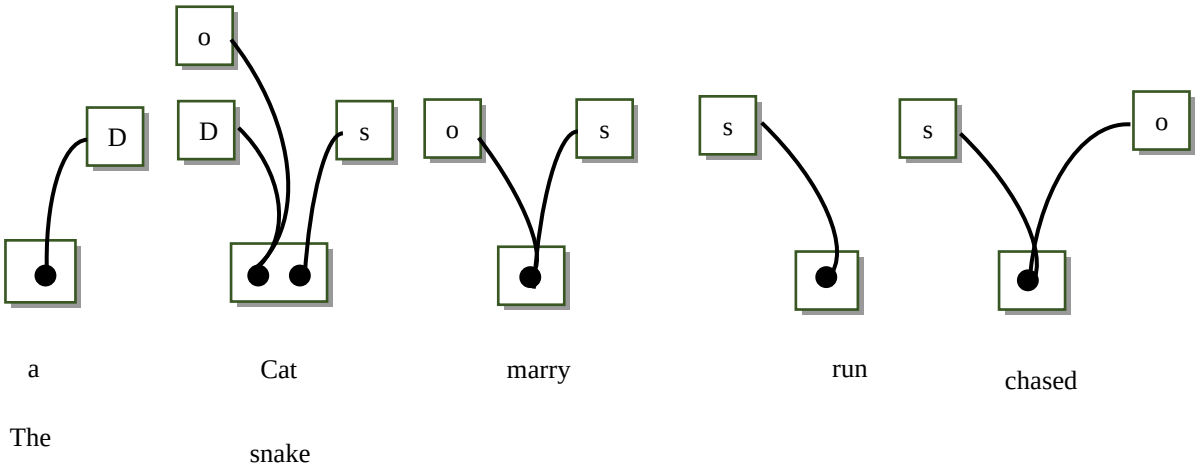


Figure 7. Linking Requirement Satisfaction

The unused connectors remain the same. Few more sentences are acceptable by the above dictionary such as 'A snake chased Mary', 'Mary ran' etc. Thus, the series of words: 'the Mary chased cat' is not acceptable by above dictionary shown in this Figure as its violets any three-specific rules.

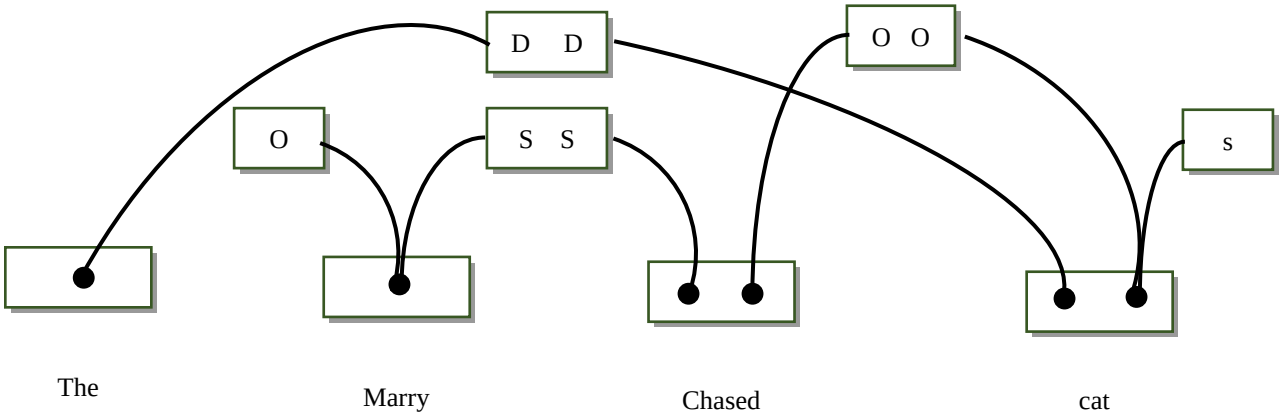


Figure 8. Linking Requirement Violation

Received: June 23, 2020 Revised: August 22, 2020 Accepted: October 16, 2020

DOI: 10.5281/zenodo.4099751

Sequence together.

Satisfaction - Links must meet the linking requirements for each word in Word sequence.

A set of links that connected between series of words in link grammar is called linkage. The link grammar dictionary consists of a group of entries, each of which defines the linking requirements of one or more words. In a dictionary, the linking requirements are specified of a formula of connectors merged by the binary operators such as '&' and 'or'. The connector is simply a character string ending in '+' or '-'. This Table describes the dictionary entries [17].

Words	Formulas
A , the	D+
Snake , cat	D- &(O- or S+)
Mary	O- or S+
Ran	S-
Chased	S- or O+
Dictionary Entry	

Equations

$$\underset{y' \in Y(x)}{\text{Argmax}}\ P(x, y')$$

Where, y(x) ---->a set of all possible grammatical structure x.

p (x, y′) ---->probability of (x, y′)

P (x, y) ---->learning of (x, y).

$P(x, y) = \prod_{r \in R(x, y)} p(r)$

R (x, y) ----->a set of production in tree (x, y).

3.3.3 Tree bank grammar:

The grammar is a series of rules that happens in the annotated corpus. There are about 17500 grammar rules in the tree bank grammar. But the major use of the Treebank is to provide the probabilities to report the statistical parsers, and the grammar does not actually have to be created (generated). Before you can parse the sentence, you need to a grammar [18].

So where do grammars come from!?

Semi-automatic groups of parse trees the sentences in some corpus (manually corrected by human annotators). In a generic lowest common denominator formalism (there is no interest to any modern linguist but representing phrases of string in actual use) [19].

METHODOLOGY

The three forms of Treebank annotation, POS tagging, syntactic bracketing, and dis-fluency annotation, are all generated by the same two-step approach, automatic annotation followed by manual correction.

Equations

$$\underset{y' \in Y(x)}{\text{Argmax}}\ P(x, y')$$

Received: June 23, 2020 Revised: August 22, 2020 Accepted: October 16, 2020

DOI: 10.5281/zenodo.4099751

Where, y(x) ----> a set of all possible grammatical structure x.
P (x, y') ---->probability of (x, y')
P (x, y) ---->learning of (x, y).
P (x, y) = ∏ r∈R (x, y) p(r)
R (x, y) ----->a set of production in tree (x, y).

☆ **The main reason for choosing Context-Free Grammar (CFG)**

1) It provides the best mathematical formula that separates a certain kind of language.
2) It can be maintained easier.
3) It possible through writing a program that determines the sentence is grammatical or not.
4) Easy to write and understand.
5) Enable simple writing of sentences by tree.

5. Conclusion

Plagiarism is the incorrect use of a source material, whether intentionally or not. More specifically, it is the non-referential use of words, ideas, and facts taken from the literature. Plagiarism occurs when an author declares that the thoughts and creativity of others are his own. Therefore, in this survey we explained the type of plagiarism and provide different software to solve plagiarism problems and discussed the process of plagiarism using NLP (natural language processing).we discussed the NLP stages in plagiarism system (pre-processing) and algorithms that use to detect plagiarism such as Lexical Analysis Algorithm, Semantic Analysis Algorithm and Grammar Analysis Algorithm. We implement these algorithms to build system detect the plagiarism from (internet sources or publications or student papers).

References

[1] Indurkhya, Nitin, and Frederick J. Damerau. Handbook of Natural Language Processing. Chapman & Hall/CRC, 2010.
[2] "Natural Language Toolkit¶." *Natural Language Toolkit - NLTK 3.5 Documentation*, www.nltk.org/.
[3] Angry Ronald Adam & Suharjito, Plagiarism Detection Algorithm using natural language processing based on grammar analyzing.
[4] Thomas Hofmann, Probabilistic Latent Semantic Analysis.
[5] Yan, Tingxu & Maxwell, Tamsin & Song, Dawei & Hou, Yuexian & Zhang, Peng. (2010).Event-Based Hyperspace Analogue to Language for Query Expansion.
[6] Azzopardi, Leif & Girolami, Mark & Crowe, Malcolm. (2005). Probabilistic hyperspace analogue to language.
[7] (n.d.).Retrieved from https://www.cs.rochester.edu/~nelson/courses/csc_173/grammars/cfg.html
[8] Context Free Grammars. (n.d.). Retrieved from https://brilliant.org/wiki/context-free grammars/#:~:text=A context-free grammar_ is,, compiler design, and linguistics.
[9] Libretexts. (2020, May 18). 4.1: Context-free Grammars. Retrieved from https://eng.libretexts.org/Bookshelves/Computer_Science/Book:_Foundations_of_Computation_(Critchlow_and_Eck)/04:_Grammars/4.01:_Context-free_Grammars
[10] Context-Free Grammar Introduction. (n.d.). Retrieved from https://www.tutorialspoint.com/automata_theory/context_free_grammar_introduction.htm
[11] CFG Simplification. (n.d.). Retrieved from https://www.tutorialspoint.com/automata_theory/cfg_simplification.htm
[12] Parsing English with a Link Grammar - arXiv. (n.d.). Retrieved from https://arxiv.org/pdf/cmp-lg/9508004v1.pdf
[13] Guest. (n.d.). A Robust Parsing Algorithm for Link Grammars. Retrieved from https://mafiadoc.com/a-robust-parsing-algorithm-for-link grammars_5b722d8b097c47f2548b457c.html
[14] Dependency Grammar and Dependency Parsing (Joakim Nivre). Retrieved from https://cl.lingfil.uu.se/~nivre/docs/05133.pdf
[15] Dependency Treebanks :Methods, Annotation Schemes and Tools(Tuomo Kakkonen).Retrieved from https://www.researchgate.net/publication/1960118_Dependency_Treebanks_Methods_Annotation_Schemesand_Tools
[16] Dependency parser (Hays 1962).retrived from https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1162/handouts/SLoSP-2014-4-dependencies.pdf

Received: June 23, 2020 Revised: August 22, 2020 Accepted: October 16, 2020
DOI: 10.5281/zenodo.4099751