https://doi.org/10.31449/inf.v42i3.1559

Using Semantic Perimeters with Ontologies to Evaluate the Semantic Similarity of Scientific Papers

Samia Iltache
Department of Computer Science, Mouloud Mammeri University, Tizi ouzou, Algeria
E-mail: siltache@gmail.com

Catherine Comparot
IRIT, Université de Toulouse, CNRS, INPT, UPS, UT1, UT2J, France
E-mail: Catherine.Comparot@irit.fr

Malik Si Mohammed
Department of Computer Science, Mouloud Mammeri University, Tizi ouzou, Algeria
E-mail: m_si_mohammed@esi.dz

Pierre-Jean Charrel
IRIT, Université de Toulouse, CNRS, INPT, UPS, UT1, UT2J, France
E-mail: Charrel@univ-tlse2.fr

Keywords: domain ontologies, semantic annotation, classification, conceptual graph, semantic perimeter, text similarity

The work presented in this paper deals with the use of ontologies to compare scientific texts. It particularly deals with scientific papers, specifically their abstracts, short texts that are relatively well structured and normally provide enough knowledge to allow a community of readers to assess the content of the associated scientific papers. The problem is, therefore, to determine how to assess the semantic proximity/similarity of two papers by examining their respective abstracts. Given that a domain ontology provides a useful way to represent knowledge relative to a given domain, this work considers ontologies relative to scientific domains. Our process begins by defining the relevant domain for an abstract through an automatic classification that makes it possible to associate this abstract to its relevant scientific domain, chosen from several candidate domains. The content of an abstract is represented in the form of a conceptual graph which is enriched to construct its semantic perimeter. As presented below, this notion of semantic perimeter usefully allows us to assess the similarity between the texts by matching their graphs. Detecting plagiarism is the main application field addressed in this paper, among the many possible application fields of our approach.

Povzetek: Prispevek obravnava uporabo ontologij za primerjavo znanstvenih besedil. Poglavitna uporaba je odkrivanje plagiacije.

1 Introduction

Assessing query-text or text-text similarity is the concern of several research domains such as information retrieval and automatic classification of documents. For many works, a document is represented by a vector of words. The very large size of the vectors reduces the effectiveness of these approaches and often requires reducing the number of dimensions to represent the document vectors. Some approaches are based on a learning corpus to compute the similarity between texts, as is done in the field of document classification. However, a large text corpus may not always be available and the result of the document classification depends and varies according to the chosen learning corpus. The similarity is based on the morphological comparison of the terms composing the query and the documents. The polysemy and synonymy inherent in the presence of certain terms of the language as well as the links between the terms are ignored, which generates erroneous matching.

In this paper, an approach to assess the similarity between texts is presented, focusing on the similarity of scientific abstracts. This approach is based on a semantic classification of documents using domain ontologies which provides a more stable base than a learning corpus. A document is no longer represented by a set of characteristics independent of each other, but by a conceptual graph extracted from the ontology to which the document is attached. The similarity between two documents is evaluated by comparing their respective graphs.

One of our propositions is to refine this process of semantic comparison through a generic structuring of an abstract of a scientific paper into distinct parts whose descriptive roles are different. The global similarity of

two abstracts will indeed be different according to whether one compares, for example, the contribution or the context of the paper, both evoked in the abstract. The proposed process constitutes a solution that can answer many problems requiring semantic comparison, as is the case, for example, in Semantic Information Retrieval. Finally, the relevance of our approach is examined by using it to highlight risks of plagiarism (expressing identical ideas using different terms), or even self-plagiarism (identical results published more than once by their authors, voluntarily using different terms).

In addition to an original process to compare the abstracts of scientific papers based on domain ontologies, and combine a classification process with a semantic comparison of conceptual graphs, one of our main contributions is the introduction of the concept of semantic perimeter which is obtained by an ontology enrichment process. The semantic perimeter plays an important role in semantic comparison as shown by our results. Our approach also introduces the possibility of structuring scientific abstracts in three distinctive parts, generally respected by authors, namely *Context*, *Contribution* and *Application domain*. Finally, this constitutes a complete process for semantic text comparison, starting by using domain ontologies, and reaching text similarity.

Section 2 of this paper covers some work related to our problematic. Section 3 describes the different steps of our text classification and comparison process and explains how to perform this process using scientific abstracts. Finally, Section 4 presents the experimentation results of our process, followed by a conclusion on the interest of such an approach and its applicability on several domains, such as giving a useful approach to constituting a documentary fund on a given knowledge domain by collecting relevant papers, which is more powerful than a mere keyword-based approach, or detecting plagiarism, which is our main purpose here.

2 Related work

2.1 Word similarity

Similarity measures are necessary for various applications in natural language processing such as word sense disambiguation [1] and automatic thesauri extraction [2]. They are also used in Web related tasks such as automatic annotation of Web pages [3]. Two classes of approaches dealing with word similarity measure can be distinguished.

Distributional approaches [4] consider a word based on its context of appearance. Words are represented by a vector of words that co-occur with them. Latent Semantic Indexation [5] is a vectorial approach that exploits co-occurrences between words. It reduces the space of words by grouping co-occurring words in the same dimensions using Singular Value Decomposition. The textual content of Wikipedia [6][7] and the Neural networks [8][9] are used for distributional word similarity to define the context of a word. In the second category, the similarity of two words is based on the similarity of their closest senses. For this purpose, a lexical resource is used, such as WordNet and MeSH. The nodes at these resources represent the meaning of the words. Measures that make it possible to calculate the degree of proximity (distance) between two nodes have been defined. Several approaches can be identified for calculating of such distances: Approaches based only on the hierarchical structure of the resource [10][11][12][13]. The measure proposed in [11] is based on edge counting and the measure proposed in [12] is based on the notion of least common super-concept; that is, the common parent of two nodes, the furthest from the root. In [13], the proposed measure takes into account the minimum distance between two nodes to their most specific common parent (*cp*) and the distance between *cp* and the root. Some approaches include information other than the hierarchical structure information, such as statistics on nodes or the informative content of nodes. To represent information content value, probabilities based on word occurrences in a given corpus are associated with each concept in the taxonomy [14][15]. Resources, such as Wikipedia [16][17] and Wiktionary [18], are also used in measuring word similarity.

2.2 Text similarity

The purpose of calculating text similarity is to identify documents with similar or different content. The different approaches dealing with textual similarity can be classified into three categories: approaches based on vector representation of document content, approaches applying text alignment, and approaches based on a graphical representation of documents and queries. Some approaches relating to each category are cited below.

2.2.1 Vector similarity

A text (document or query) is projected into a vector space where each dimension is represented by an indexing term. Each element of a vector consists of a weight associated with an indexing term. This weight represents the importance of a term and is calculated on the basis of TF-IDF [19] or its variants. The vector similarity is computed using several metrics such as the cosine measurement which measures the cosine of the angle formed by the vectors corresponding to the texts. Two texts are similar if their vectors are close in the vector space in which they are represented.

- Document retrieval

The vector model is proposed by Salton in the SMART system [20]. To retrieve the documents that best meet a user need, a document and a query are represented by a vector. The relevance of a document to a query is measured by a similarity based on the distance between their respective vector. Adaptations of the basic model have been proposed for processing structured documents [21][22]. The Extended Vector Space Model is one of the first adaptations of the vector model proposed by Fox [22]. A document is represented by an extended vector containing different information classes referred to as objective identifiers (denoted by c-type) such as author,

title and bibliographic references. The similarity between a document *d* and a query *q* is computed by a measure of similarity which is a linear combination of the different sub vector similarities.

Conventional Information Retrieval considers documents only based on their textual content. The evolution of the document content towards a structured representation and more precisely towards the XML format raises new issues. In [23], the author presents a Searching XML documents through xml fragments. A fragment is a text delimited by a structure. The queries are transformed into XML fragments and, for each document, a profile is created. This profile is represented by a vector composed of the pairs (t, c), where *c* is the context of appearance of the term *t*. The context is assimilated to the element with its path. An entry in the index is no longer a term but a pair (t, c). Another adaptation of the vector model described in [24] based on the computation of the cosine makes it possible to compute the similarity between a node *n*, belonging to a tree representing a document, and a query *q*. In [25], the corpus is represented by a labeled tree where each sub-tree is considered as a logical document. The authors introduce the notion of structural term (*s-term*) which is a labeled tree. An *s-term* may be an element, an attribute, or a term. The similarity between a query and a document is computed by the scalar product of the vectors. The weight of the terms is computed during the retrieval phase since the notion of logical tree is defined according to the structure of the query.

- Document classification.

Automatic texts classification makes it possible to group documents dealing with similar themes around the same class. Supervised classification approaches assign documents to predefined classes [26][27][28] while unsupervised classification approaches automatically define classes, referred to as clusters, [29].

In the supervised classification, classifiers use two document collections: A collection containing training documents to determine the characteristics of each category and a collection containing new documents to be automatically classified. The classification of a new document depends on the characteristics selected for each category. There are various supervised machine learning classification techniques. In [30], the author provides a comparison of their features.

The method based on the K Nearest Neighbors (KNN) [28][31] assumes that if the vectorial representations of two documents are close in vector space, they have a strong probability of belonging to the same category. A new document *d* is compared with documents belonging to the training set. The category assigned to document *d* depends on the category of its K nearest neighboring documents. To determine the category to be assigned to the document *d*, the most assigned class to the K neighbors closest to *d* is chosen or a weight is assigned to the different classes of k nearest neighbors according to the classification of these neighbors. Thus the class with the highest weight will be retained.

With Support Vector Machines (SVM), documents are represented in a vector space by the indexing terms that compose them. Using a training phase, this method defines a separating surface, called hyperplan, between the documents belonging to two classes which maximize the distance between this hyperplan and the nearest documents and minimizes categorization errors [32]. A category *c* is assigned to a new document *d* as a function of the position of *d* relative to the separating surface.

Some classifiers create a "prototype" class from the training collection [26]. This class is represented by the mean vector of all the document vectors in the collection. Only some features are retained which constitutes a loss of information. Some approaches replace the training collection with data extracted from "world knowledge" such as Open Directory Project (ODP) [33]. Other approaches exploit thesauri or domain ontologies with conventional classifiers (SVM, Naive Bayes, K-means, etc.) and represent a document by a vector whose features are concepts or a set of terms and concepts [29][34][35].

As reported in [36], approaches using the vector representation of documents have several limitations: Their performances decrease as soon as they apply to relatively long texts. With the weighting formulas used, words appearing only once in the document or, on the contrary, words that are often repeated are ignored although they have a meaning with respect to the content of the document. The vector representation as defined does not highlight the relationships between words in a document, thus generating erroneous matching.

A document is represented by a vector whose size is equal to the number of features retained to represent the various categories, in the case of classification, and the number of terms used to represent the corpus, in the case of information retrieval. In [37], the authors studied the impact of the number of dimensions on the "nearest neighbor" problem. Their analysis revealed that when this number increases, the distance to the nearest data point approaches the distance to the farthest data point.

2.2.2 Sentence alignment

Approaches dealing with sentence alignment are divided into three categories. Syntactic approaches based on morphological word comparison, semantic approaches using sentence structure and approaches that combine syntax and semantics. Gunasinghe [38] proposes a hybrid algorithm that combines syntactic and semantic similarity and uses a vectorial representation of sentences by using WordNet. This algorithm takes into account two types of relationship in the sentence pairs: relationships between verbs and relationships between nouns. Liu [39] proposes an approach to evaluate the semantic similarity between two sentences. They use a regression model, Support Vector Regression, combined with features defined using WordNet, corpus, alignment and other features to cover various aspects of sentences. Other approaches perform the text alignment by comparing all the words preserving their order in sentences. However, these algorithms are rather slow and they do not

dissociate terms describing the theme of the document from those used to build sentences. In [40], authors use a text alignment algorithm [41] to align a text with the set of documents in a corpus. This algorithm uses a matrix in which the deletion or insertion of a word is represented by -1, a mismatch by a 0 while a match is represented by its IDF weight. The authors use a full-text alignment where the highest score from any cell in the alignment matrix represents the similarity score of two texts. In [42], authors introduce a new type of sentence similarity called Structural Similarity for informal, social network styled sentences. Their approach eliminates syntactic and grammatical features and performs a disambiguation process without syntactic parsing or POS Tagging. They focus on sentence structures to discover purpose- or emotion-level similarities between sentences.

2.2.3 Graph similarity

Assessing of the graph similarity is used, in particular, in the field of Information Retrieval. The document and query are both represented by a conceptual graph constructed from a domain ontology or a thesaurus.

In the domain of Semantic Information Retrieval, Dudognon [43] represents the documents by a set of "annotations". Each annotation consists of several conceptual graphs. The similarity between two graphs is defined as the weighted average of the similarities between the concepts that compose this graphs and the similarity between two "annotations" is computed by the mean of similarities of their conceptual graphs. Baziz [44] suggests constructing a graph for each document and for each query using concepts extracted from WordNet. A mapping of the graph of a document to that of the query leads the author to represent the two graphs with respect to the same reference graph made up of nodes belonging to the document and to the query. Each graph is then expanded by adding nodes of the reference graph. The weights of the nodes added to the query are zero whereas in the sub-tree of the document where a node is added, the weight of a level s node is updated recursively by multiplying the weight of the level $s + 1$ node (the level s node subsumes the level $s + 1$ node) by a factor which depends on the hierarchy level. The two representations are then compared using fuzzy operators and a relevance value is computed. This value expresses the extent to which the document covers the subject expressed in the query. Shenoy [45] represents a document by a "sub-ontology" constructed using the demo version of ONTO GEN Ontology Learner which is part of the TAO Project. Two documents are compared by applying the alignment of their "sub-ontology" based on the number of concepts, properties and relationships contained in each document. In [46], the authors propose a unified framework of graph-based text similarity measurement by using Wikipedia as background knowledge. They call each article in Wikipedia a Wikipedia concept. For each document, the authors extract representative keywords or phrases and then map them into Wikipedia concepts. These concepts constitute the nodes at the bottom of the bipartite graph. There is an edge between a document node and a concept node if the concept appears in the specific document. The weight of the edge is determined by the frequency of the concept's occurrence in that document. The similarity of two documents is determined by the similarity of the concepts they contain. The authors in [18] present a unified graph-based approach for measuring semantic similarity between linguistic items at multiple levels: senses, words, and sentences. The authors construct different semantic networks. One of them is based on WordNet. The nodes in the WordNet semantic network represent individual concepts, while edges denote manually-crafted concept-to-concept relations. This graph is enriched by connecting a sense with all the other senses that appear in its disambiguated gloss. Measuring the semantic similarity of a pair of linguistic items consists of an Alignment-based Disambiguation and a random Walk on a semantic network. In [47], authors propose a graph-based text representation, which is capable of capturing term order, term frequency, term co-occurrence, and term context in documents. A document is represented by a graph. A node represents a concept: a set of single word or phrase and an edge is constructed based on proximity and co-occurrence relationship between concepts. In addition; the associations among concepts are represented through their contexts. The nodes within the window (e.g. paragraph, sentence) are linked by weighted bidirectional edges. The approach described in [48] presents a graph-based method to select the related keywords for short text enrichment. This method exploits topics as background knowledge. The authors extract topics and re-rank the keywords distribution under each topic according to an improved TF-IDF-like score. Then, a topic-keyword graph is constructed to prepare for link analysis. In [49], the authors create a semantic representation of a collection of text documents and propose an algorithm to connect them into a graph. Each node in a graph corresponds to a document and contains a subset of document words. The authors define a feature and document similarity measures based on the distance between the features in the graph.

2.3 Detecting plagiarism

Plagiarism consists in copying a work of an author and presenting it as one's own original work. Plagiarism detection systems usually have the original document and the suspicious document as inputs. They focus on the following points: an exact copy of the text (copy/paste), inserting or deleting words, substituting words (use of synonyms), reformulation and modification of sentences structure. In n-gram approach, a text is characterized by sequences of n consecutive characters [50][51][52]. Based on statistical measures, each document can be described with so called fingerprints, where n-grams are hashed and then selected to be fingerprints [53]. An overlap of two fingerprints extracted from the suspicious and source documents indicates a possibly plagiarized text passage. Statistical methods [54] do not require an understanding of the meaning of the documents. The

common approach is to construct the document vector from values describing the document such as the frequency of terms. Comparing the source document with the suspicious document, amounts to calculating their degree of similarity on the basis of different measures (BM25, language model, etc.). Vani [55] segments the source document and the suspicious document into sentences. Each sentence is then represented by a vector of weighted terms that compose it. Each sentence of the source document is compared to all the sentences of the suspicious document and similarity between two vectors is computed using, individually, several metrics (cosine, dice, etc.). Vani studies the importance of the combination of these various metrics on detecting plagiarism. He also explores the impact of the use of POS Tagging on calculating of sentence similarity. The sentences labeled by a syntactic parser are thus compared by matching the terms belonging to the same class (nouns with nouns, verbs with verbs, adjectives with adjectives and adverbs with adverbs). Other approaches based on sentences alignment compute the overlapping percentage of words or sentences between the source document and the suspicious document. These methods do not permit the detection of cases of plagiarism where synonymy is used to replace words in the reformulation of sentences. The representation of a document by a graph is also used in detecting plagiarism. In [45], the alignment of "sub-ontologies" is based on the number of concepts, properties and relations corresponding to the original document and the suspicious document. Alignment is expressed as a fraction of the whole. If this fraction is above a given threshold, the system concludes that the two documents are similar in meaning. Osman [56] describes an approach of detecting plagiarism by representing documents (original and suspicious) with a graph deduced from WordNet. This approach is useful in detecting forms of plagiarism where synonymy is used to reformulate sentences. The document is divided into sentences. Each node of the graph constructed for the document represents the terms of a sentence. The terms of sentences are projected on WordNet to extract the concepts corresponding to them. Each relationship between two nodes is represented by the overlap between the concepts of the two nodes. These concepts help in detecting suspicious parts of a document.

An important characteristic of our approach lies in the fact that it is not necessary to have a reference document a priori, since any document can be compared with a corpus dealing with the same knowledge domain as identified in the first step of our process that is proposed here.

3 Proposed approach

The representation of a document by a semantic graph is used in different domains such as information retrieval [43][44], plagiarism detection [45][56] and document summarization [57]. However, these graphs differ in the way they are constructed. The purpose of our approach is to assess the semantic similarity between textual documents. Unlike conventional approaches, a document is not represented by a vector. Our approach is to build a conceptual representation of a text in the form of a semantic graph in which the nodes and arcs correspond respectively to concepts and relationships between concepts extracted from the domain ontology chosen.

The similarity between two texts is evaluated in two steps. The first step is to perform a semantic classification of documents based on domain ontologies. The classification makes it possible to deduce an overall similarity defined by the context in which the content of the document is used. The second step compares and evaluates the similarity of two texts related to the same domain ontology by comparing their constructed and enriched graph as explained in the following sections.

3.1 Classification of documents

The process is based on a semantic classification of texts using domain ontologies [58]. Figure 1 summarizes the classification process.

The classification groups documents according to the knowledge domain covered by their content. This grouping identifies an overall similarity and involves several steps.

- *Projection, extraction of terms and candidate concepts*. The "projection" of a document on different ontologies helps to associate meaning to the terms of the document with respect to concepts belonging to these ontologies and to select the candidate concepts. The notion of concept gives a meaning to a term relative to the domain in which this concept is defined. The whole document is divided into sentences. Each sentence is browsed from left to right from the first word. The words of each sentence are projected, before pruning stop words, on different domain ontologies to extract longer phrases (groups of adjacent words in a sentence called "terms") that denote concepts. This choice is determined by: 1) the concepts are often represented by labels consisting of several words. An example of mono- and multi-word concepts is given in table 1. 2) long terms are less ambiguous and better determine the meaning conveyed by the sentence. Several concepts belonging to the same domain ontology may be candidates for a given term. The following example shows to what extent it is important to bring out the longest terms and the longest concept.

For the sentence: "The **Secretary of State for the Home Department** had clearly indicated that evidence obtained by torture was inadmissible in any legal proceedings," the synsets in Table 1 are extracted from WordNet.

As shown in Table 1, there are several synsets in WordNet that correspond to the words "secretary of state for the home department" in the sentence. These synsets have one or more words.

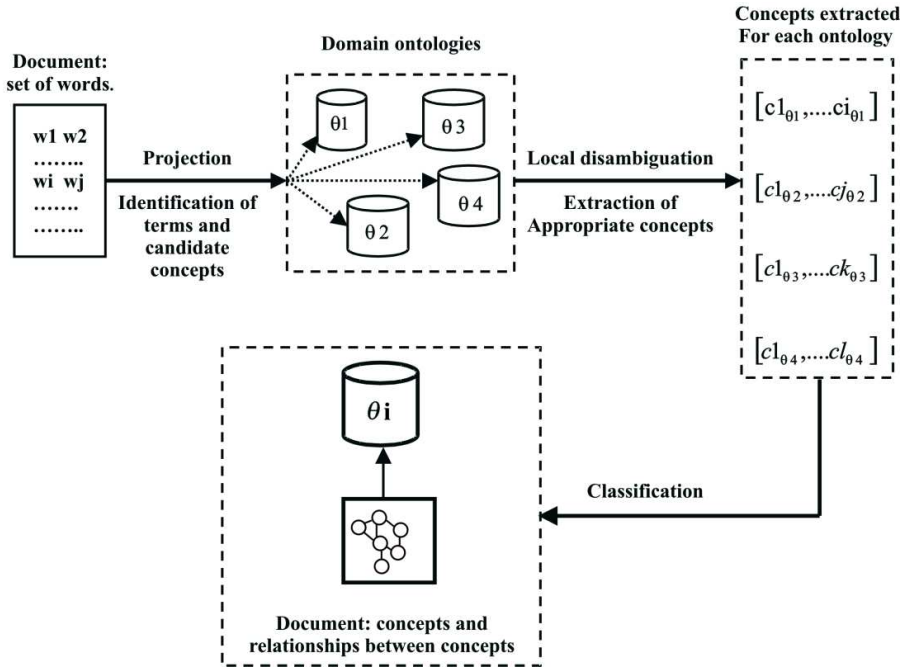

Figure 1: Classification of a document.

Words in a sentence	Synset label in WordNet	N° synset in WordNet
Secretary of State for the home Department	secretary_of_state_for_the_home_department	**09526473**
	secretary_of_state	09883412 09455599 00569400
	secretary	09880743 09880504 09836400 04007053
	state	07682724 08125703 07673557 00024568 07646257 08023668 13192180 13656873
	home	08037383 03141215 07973910 13687178 03398332 07974113 07587703 03399133 08060597
	department	07623945 08027411 05514261

Table 1: Extraction of terms and synsets.

The longest term "secretary of state for the home department" is extracted from the sentence. It corresponds to the synset secretary_of_state_for_the_ home_department (09526473), which represents the correct sense in the sentence.

- *Local disambiguation.* In the projection step, for each ontology, all the candidate concepts for a given term are extracted. The local disambiguation process is used to select for a term *t* the most appropriate concept among several candidates belonging to the same ontology. To do this, the context of occurrence of the term *t* in the document is taken into consideration.

The appropriate concept for the term *t* is chosen, taking into account both the semantic distance between the term *t* with neighboring terms, (i.e., which occur in its context), and the semantic distance between concepts associated with the term *t* and concepts corresponding to the neighboring terms in the ontology considered.

The meaning of a term *t* in a document is determined by its nearest unambiguous neighbors terms. *t* will then be disambiguated by its nearest neighbor on the left or by its nearest neighbor on the right. In case the left and right neighbors exist simultaneously, they will both be taken into consideration.

The disambiguation process is then done at three levels, starting at the sentence level. For each sentence, the ambiguous terms are disambiguated considering their left and right neighbors in the sentence. Any disambiguated term helps to move forward in the process of disambiguation of next terms. This process is repeated in case ambiguous terms still remain, considering in a second step the paragraph level, and finally, if necessary, the document level. The local disambiguation process at the sentence level, summarized by the algorithm in Figure 2, considers neighboring terms, unambiguous, that have associated concepts in the ontology considered, surrounding *t*: it retrieves the concepts *Cnl* and *Cnr*, corresponding respectively to *nl*, the nearest neighbor on the left of *t* and *nr*, the nearest neighbor on the right of *t*. The appropriate concept for the term *t* among candidate concepts is the semantically nearest concept of *Cnl* or

Cnr. This amounts to browsing the ontology and calculating the minimum distance between each concept associated with t and candidate concepts Cnl, Cnr.

Several existing metrics in the literature are used to calculate this minimum distance. An example of local disambiguation in the domain anatomy of WordNet is given in the Figure3.

```
Input
    Ec = {extracted concepts for S}  {S, current sentence}
    Et = {terms belonging to S}
    E = {Unambiguous terms of S}
Output
   Ec = {retained concepts for S}

Procedure disambiguation (i:integer)
  var
    j:integer
Begin
  t ←S[i]
  nl (t) ← S[i-1]
  nr (t) ← S[i+1]
  if (nl (t) in E ) and (nr (t) in E)  then
    compute  Min-dist ((Ci,Cnl), (Ci,Cnr))  {Ci, The concepts associated with t}
    E ← E ∪ t
    Ec ← Ec ∪ C        {C, retained concept for t}
  else
    if (nl (t) in E)  then
      compute  Min-dist (Ci,Cnl)    {Cnl: The concepts associated with nl}
    E ← E ∪ t
    Ec ← Ec ∪ C
   else
      if (nr (t) in E) then
        compute  Min-dist (Ci,Cnr)    {Cnr: The concepts associated with nr}
        E ← E ∪ t
        Ec ← Ec ∪ C
      else
        j ← i + 1
        disambiguation(j)
        pos ← pos + 1
        t ← S[j-1]
        compute  Min-dist (Cj-1,Cnr)
        E ← E ∪ t
        Ec ← Ec ∪ C
       End if
     End if
  End if
End
Begin
  Ec←Ø
  pos ← 1
  k ← 1
  t ← S[k]
  while ( not end (S) ) do
    if ( t not in E)  then
      disambiguation (k)
       k ← pos + 1
       pos ← pos + 1
    else
      Ec ← Ec ∪ C
      pos← k+1
      k ←k + 1
    end if
    t ← S[k]
  end while
 end.
```

Figure 2: Local disambiguation at the sentence level.

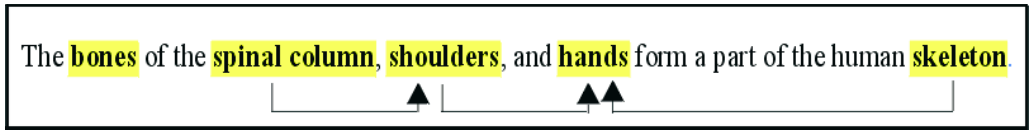

Figure 3: Disambiguation of shoulder and hand.

Table 2 shows the terms and their senses (synsets) in the domain *anatomy* of WordNet. The different calculated distances help in choosing the most appropriate synset for each ambiguous term.

The term *shoulder* in the sentence is ambiguous. To disambiguate it, *spinal column*, its nearest unambiguous neighbor term on the left, is considered. The synset retained is 05231159.

The term *hand* in the sentence is ambiguous. Its disambiguation is done using *shoulder* and *skeleton*, its two nearest unambiguous neighboring terms on the left and right. The synset retained is 05246212.

Words in a sentence	Synset label (Anatomy)	N° synset	Distance between synsets	Terms extracted
Bones	bone	04966339		bone
	Spinal_column	05268544		Spinal column
Spinal	shoulder	**05231159**	Dist(05268544,05231159)= 0.42857143	shoulder
		05231380	Dist(05268544, 05231380)= 0.5	hand
Column				skeleton
	hand	**05246212**	Dist(05246212,05231159)= 0.42857143	
Shoulders (ambiguous)		02352577	Dist(02352577,05231159)= 0.6363636	
Hands (ambiguous)			Dist(05246212,05265883)= 0.42857143	
skeleton			Dist(02352577,05265883)= 0.6363636	
	skeleton	05265883		

Table 2: Disambiguation of ambiguous terms.

At the end of the preceding steps, a document d is represented by several sets of concepts extracted from the domain ontologies θ_i on which it was projected. These sets are represented by (1).

$$d = \begin{cases} \theta_1^d = \{c_{11}, c_{21} \ldots\ldots, c_{n1}\} \\ \theta_i^d = \{c_{1i}, c_{2i} \ldots\ldots, c_{ni}\} \\ \ldots\ldots\ldots \\ \ldots\ldots\ldots \end{cases} \quad (1)$$

- *Global disambiguation.* The classifier must be able to conclude about the relevance of a document relative to a given context and to choose from the different ontological representations the one that best corresponds to its context. A score is calculated for each document. The highest score determines the candidate ontology to be selected to represent document d.

The different terms in a document, taken together considering the contextual relations linking them, make it possible to conduct a semantic evaluation of the textual content. A matrix, defined by (2), is associated for each ontology and for each document.

$$M_{\theta_i}^d = \begin{pmatrix} lc_1c_1 & lc_1c_2 \ldots\ldots & lc_1c_n \\ & & \\ lc_nc_1 & lc_nc_2 \ldots\ldots & lc_nc_n \end{pmatrix} \quad (2)$$

The rows and columns of this matrix represent all the concepts extracted from the ontology θ_i for the document d.

C_i is the selected concept for the term t_i after projection of the document d on θ_i and lc_ic_j represents the weight of the link between the concept C_i and the concept C_j (i≠j).

The matrix is initialized to zero.

If a term t_i and a term t_j appear together within the same paragraph of the document d and the concepts Ci and Cj respectively correspond to the terms t_i and t_j, then the weight $lcicj = 1$.

The weight $lcicj$ is updated whenever the terms t_i and t_j appear together in the same paragraph.

The weight $lcici$ corresponds to the appearance of the term t_i in the document. It is equal to 1.

The weight $lcicj$ is updated for all paragraphs of the document d.

The importance of the concept C_i in document d is determined by its total weight in d relatively to the ontology θ_i. This weight is given by the row associated with it in the matrix.

The score for each ontology obtained from the sum of the weights of all concepts extracted from this ontology for the document d measures the extent to which each ontology represents this document. The ontology that gets the highest score will be selected to represent the document d.

For documents belonging to the same knowledge domain, their "local" semantic similarity is computed.

The process compares their content using their semantic perimeter – a notion that is introduced and defined later in the paper – constructed on the basis of their conceptual graph extracted from the ontology to which they are attached.

3.2 Text similarity and semantic perimeter

An author describes the subject of his document by evoking one or more different notions. He can describe them by addressing several sub-notions. These notions and/or sub-notions can be described in a general or precise way according to the level of detail to be highlighted.

In an ontology, there exists a certain structure defining the meaning of information representing a given knowledge domain and the way in which this information is related to each other. This structure is defined by several branches representing different hierarchies. Each hierarchy has branches to separate data with common characteristics but also different characteristics. The tree of Figure 4, inspired by the *geometric figures* ontology proposed in [59], shows two branches *Br1* (*figure)* and *Br2* (*angle*) representing two different data. Branch *Br2* has two sub-branches 2.1 and 2.2 corresponding respectively to a *right angle* and an *acute angle*. *Right angle* and *acute angle* are two concepts with different characteristics but common characteristics defined by their common parent a*ngle*.

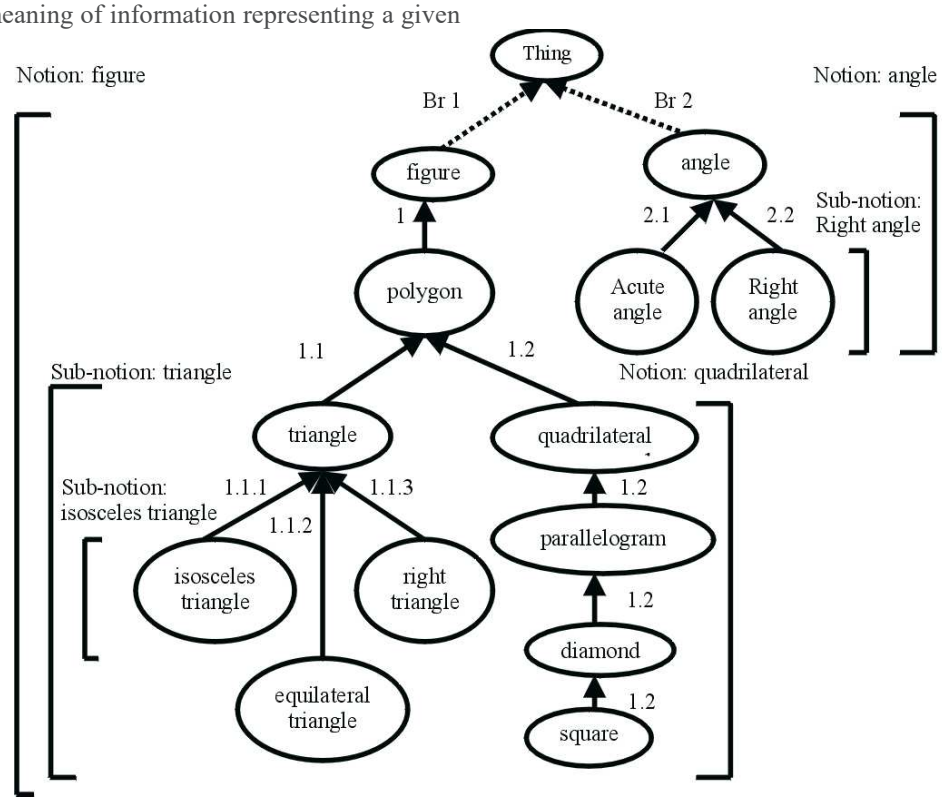

Figure 4: Extract from the *geometric figures* ontology.

3.2.1 Objective of the approach

Consider two texts *Txt1* and *Txt2,* previously classified in the same knowledge domain represented by a domain ontology, whose similarity needs to be assessed: *Sim (Txt1, Txt2*). Our semantic similarity process is based on the following assumptions:

1 Each branch/sub-branch of the ontology is associated with a notion/sub-notion described in a document.

2 Concepts linked by "*is-a*" relations form a branch.

3 A branch can have several sub-branches.

4 Two branches with the root of the ontology as the only common parent represent two different notions.

5 Two sub-branches having a common parent represent two different sub-notions sharing common characteristics defined by their common parent.

6 The weight of an *initial concept* is equal to 1.

7 The weight of an added concept representing implicit information is less than 1.

8 The similarity between two texts varies between 0 and 1.

Our approach is based on the identification of the branches to which the concepts of the documents belong and on the enrichment of the conceptual graphs of these documents. Associating a notion with a branch helps in identifying different and identical notions. It can be said for example that the notion "*angle*" is different from the

notion "*figure*" and that the notion "*triangle*" is different from the notion "*quadrilateral*" because they belong to different branches or sub-branches. The concepts *quadrilateral*, *parallelogram*, *diamond,* and *square* belong to the same sub-branch describing the same notion. Each of them brings a degree of precision knowing that this precision is increasingly higher the further one goes down the hierarchy.

Graph enrichment highlights common notions to two documents without these being explicitly cited in their content and makes it possible to deduce similarities between notions by examining the branches to which their corresponding concepts belong.

3.2.2 Graph enrichment

To describe a given subject, the authors, can choose different words and different levels of description depending on the importance that each of them wishes to give to a notion addressed in the text. Thus, by adding concepts, graph enrichment makes it possible to deduce implicit information that can be shared by these two texts.

Like Baziz [44], our process enriches the text graphs by adding concepts. The applied enrichment differs from that achieved by Baziz in the choice of concepts to be added and the weight assigned to these concepts. For our case, the weight assigned to the concepts helps in defining the implicit or explicit presence of a concept.

A graph is enriched by constructing the semantic perimeter of its corresponding text and comparing it to another graph.

3.2.2.1 Constructing the semantic perimeter of a text

Definition 1: The semantic perimeter of a text is a graph whose nodes are the initial concepts and the link concepts. Initial concepts are extracted from the domain ontology to which the document is attached. These concepts represent the information explicitly described in its content. With these concepts, a conceptual graph is constructed and enriched by link concepts representing the implicit information in the text that is deduced from the initial concepts and through browsing the "is-a" relationships and the transversal relationships defined in the domain ontology. The semantic perimeter thus constructed for each document makes it possible to evaluate their semantic similarity even if these documents describe the same ideas with different terms.

- *Constructing the graph of initial concepts*

During the classification process, a text is projected onto a set of domain ontologies. At the end of this step, the text is represented by a conceptual graph, whose nodes constitute the *initial concepts*.

These concepts correspond to the terms explicitly cited in the document.

- *Constructing the semantic perimeter*

The *link concepts* extracted from the ontology, being on the shortest path linking the *initial concepts Ci* and *Cj* by *is-a* relations or transversal relations, are added to the graph of a document.

Link concepts are selected in order to retain only concepts that make sense in relation to the knowledge domain represented by the ontology. In fact, some concepts represented in an ontology are used to construct the structure of the ontology but have no meaning for the domain in question.

Example: *host* and *hard_disk,* are two synsets extracted for a document classified in the *computer_ science* domain. Figure 5 shows the synsets linking them in WordNet.

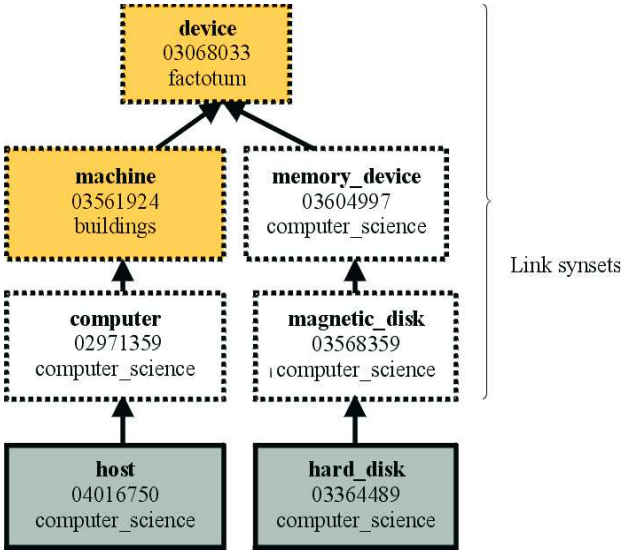

Figure 5: link synset linking host to hard_disk.

The *link synsets* are: {*computer* 02971359, *machine* 03561924, *device* 03068033, *memory_device* 03604997 and *magnetic_disk* 03568359}. The synsets *machine* 03561924 and *device* 03068033 are not retained, since they respectively belong to the *buildings* domain and *factotum* domain.

3.2.2.2 Comparing graphs

Comparing two texts *Txt1* and *Txt2* is carried out from their semantic perimeter *G1* and *G2*. A mutual enrichment of these two graphs is achieved by comparing the concepts belonging to *G1* with the concepts belonging to *G2*. Each graph enriched the other and concepts are added to *G1* and/or to *G2*. This is done by browsing the graphs from leaf nodes to the root as follow:

- If the graph *G1* (the graph *G2*) contains a concept *C1* and the graph *G2* (the graph *G1*) contains a concept *C2* such that *C2* is an ancestor of *C1*, then the concept *C2* is added to the graph *G1* (to the graph *G2*).
- The graphs are also enriched by adding the common parents to concepts belonging to graphs *G1* and *G2*. This enrichment is done in two steps:

- By considering concepts belonging only to the graph *G1* (to the graph *G2*).
- By considering the concepts belonging to graphs G1 and G2.

By adding *common parent* concepts, graph enrichment helps in determining the common branches and sub-branches to *G1* and *G2* and thus to deduce an implicit similarity between *Txt1* and *Txt2*.

As an illustration, in the *geometric figures* domain represented by figure 4, three texts (*T1*, *T2* and *T3*) are considered, and their content is as follows:

T1: *A square is a regular polygon with four sides. It has four right angles and its sides have the same measure.*

T2: *A diamond is a parallelogram. Some diamonds have four equal angles.*

T3: *A triangle has three sides. If it has a right angle, it is a right triangle.*

- Let us compare *T1* and *T2*.

The semantic perimeters of *T1* and *T2* and the comparison of their respective graphs *G1* and *G2* are given in Figure 6.

The projection of the texts *T1* and *T2* on the ontology represented by figure 4, allows us to find the initial concepts to construct graphs *G1* and *G2*.

G1 is represented by the concepts (*square*, *polygon*, *right angle*) and *G2* is represented by the concepts (*diamond*, *parallelogram* and *angle)*. At this stage, the graphs have no common concept.

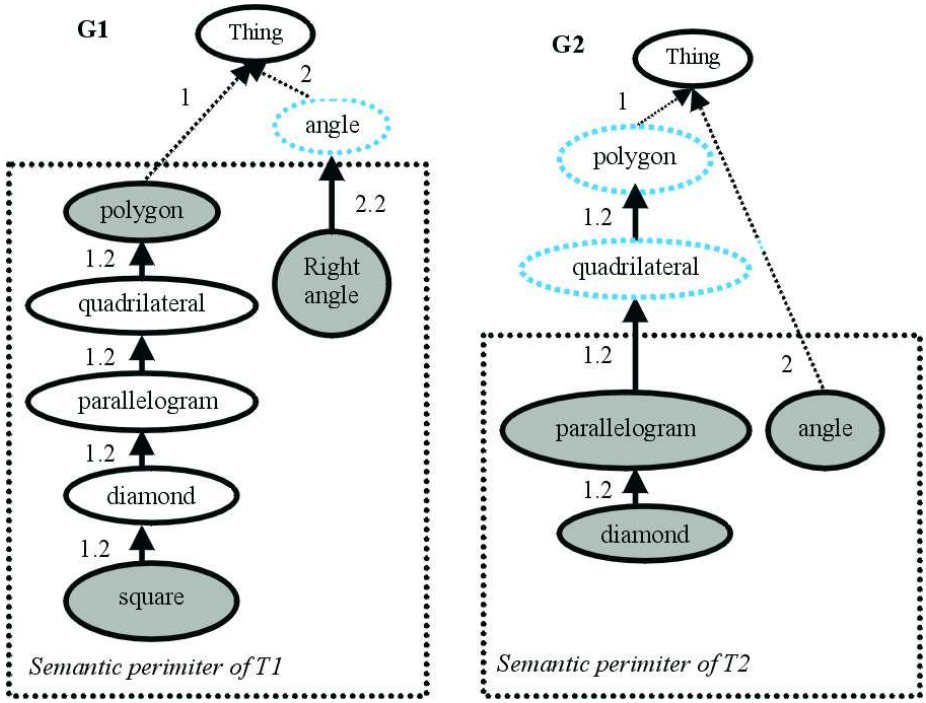

Figure 6: Comparison and enrichment of graphs corresponding to T1 and T2.

The enrichment of these two graphs made it possible to add concepts semantically linked to the initial concepts and to bring out common concepts to the two texts, not explicitly cited in their contents. The common concepts are *diamond*, *parallelogram, quadrilateral*, *polygon* and *angle*.

- Let us compare T2 and T3.

The semantic perimeters of *T2* and *T3* and the comparison of their respective graphs *G2* and *G3* are given in Figure 7.

The projection of the texts *T2* and *T3* on the ontology, represented by figure 4, allows us to find the initial concepts to construct graphs *G2* and *G3*.

G2 is represented by the concepts (*diamond*, *parallelogram* and *angle*) and *G3* is represented by the concepts (*triangle*, *right triangle* and *right angle*). The enrichment of the two graphs enabled us to find common concepts (*angle* and *polygon*).

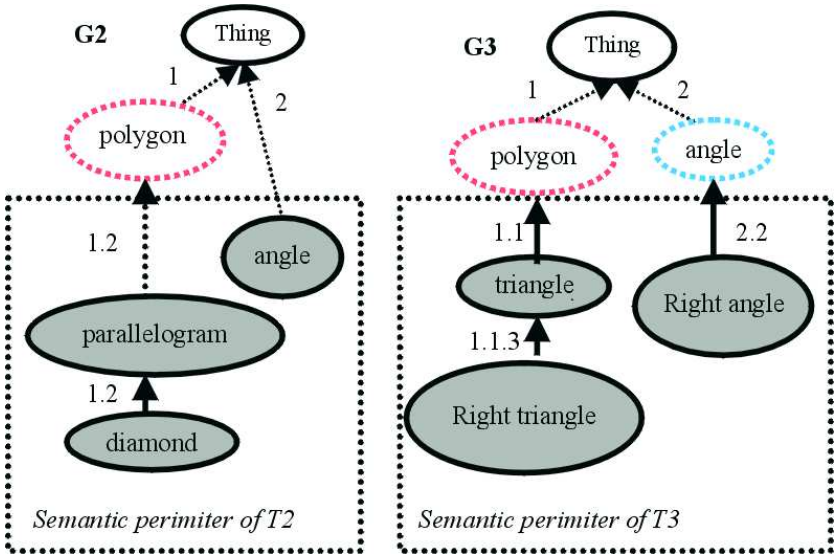

Figure 7: Comparison and enrichment of graphs corresponding to T2 and T3.

3.2.3 Calculating the similarity of two texts

Definition 2*:* Textual similarity is defined by the set of common notions and sub-notions addressed by two texts. It is a function of the concepts corresponding to these texts, their weight and the branches to which these concepts belong. The similarity of two texts is given by the similarity of their respective graphs according to equation (3).

$$Sim(Txt1, Txt2) = Sim(G_{Txt1}, G_{Txt2}) \quad (3)$$

3.2.3.1 Weight of the concepts

The weight attributed to an *initial concept* is equal to 1. This weight defines the explicit presence of the concept in the document. Concepts belonging to the same branch do not have the same semantic weight: concepts at the top of the hierarchy have a more general meaning than concepts at the bottom of the hierarchy that represent a more precise meaning. The more one descends towards the bottom of the hierarchy, the more precise the meaning of the concepts is. Thus, to a concept added to graph *G1* during the enrichment process, a weight whose value is less than 1 is assigned. This weight represents the value of an implicit information and is calculated based on parameter *g*. *g* expresses the degree of generalization of a *parent concept* vis-a-vis its *child concept*.

Like Fuhr [60] and Baziz [44], who reduce the weight of the nodes of a tree representing a document according to their position with respect to the most specific nodes by multiplying by a factor whose value is between 0 and 1, our process computes the weight of an added concept by using parameter g whose value is between 0 and 0.1 according to equation (4).

$$P(C_j) = 1 - (g \times (length(C_i, C_j)) \quad (4)$$

Cj is the added concept and *Ci* is the initial concept, belonging to *G1* and/or to *G2,* the lowest in the branch to which *Cj* is added and *Length (ci, cj)* indicates the number of arc linking *Cj* to *Ci* in the branch.

3.2.3.2 Semantic similarity of two graphs G1 and G2

A factor is introduced indicating the percentage of common notions described by two texts. Its value is calculated by the number of common branches relative to the total number of branches belonging to the two graphs. The similarity between two graphs *G1* and *G2* is computed using equation (5).

$$Sim(G1, G2) = \frac{nbBc_{(G1,G2)}}{nbB_{(G1,G2)}} \times \frac{\sum_{Bc} \sum_{Ccom \in Bc} P(C_{com})}{\sum_{B} \sum_{C \in B} P(C)} \quad (5)$$

B represents any branch belonging to the graphs while *Bc* represents a common branch to both graphs. *C* is a concept belonging to graphs *G1*, *G2* and *Ccom* is a common concept to both graphs. $nbBc_{(G1,G2)}$ and $nbB_{(G1,G2)}$ respectively represent the number of common branches and the total number of branches belonging to the two graphs.

3.2.3.3 Example

Let us again take the examples shown in Figures 6 and 7 and summarize the various results in Tables 3 and 4. For parameter *g*, the value 0.05 is used.

Initially, *G1* and *G2* showed no concept in common and, therefore, a priori no similarity. The same applies to graphs *G2* and *G3*. The enrichment of the graphs helped to bring out a similarity between *T1* and *T2*, as well as between *T2* and *T3* that is not explicitly described in their content. The results also show that text *T2* is semantically closer to *T1* than to *T3*.

Texts	Concepts	Type	Weight
T1	square	initial	1
	diamond	link	0,95
	parallelogram	link	0,90
	quadrilateral	link	0,85
	polygon	initial	1
	angle	Ancestor	0,95
	Right angle	initial	1
T2	diamond	initial	1
	parallelogram	initial	1
	quadrilateral	Ancestor	0,85
	polygon	Ancestor	0,80
	Angle	initial	1
Common branches		1 1.2 2	
All branches		1 1.2 2 2.2	

Table 3: Concepts of T1 and T2 after enriching their respective graphs.

Texts	Concepts	Type	Weight
T2	diamond	initial	1
	parallelogram	initial	1
	polygon	Common parent	0,85
	angle	initial	1
T3	right angle	initial	1
	right triangle	initial	1
	triangle	initial	1
	polygon	Common parent	0,85
	angle	ancestor	0,95
Common branches		1 2	
All branches		1 1.1 1.2 1.1.3 2 2.2	

Table 4: Concepts of T2 and T3 after enriching their respective graphs.

$$Sim(T1,T2) = \frac{3}{4} \times \frac{(0,80) + (0,85 + 0,90 + 0,95) + (0,95)}{(1) + (0,85 + 1 + 1 + 1) + (1) + (1)} = 0,49$$

$$Sim(T2,T3) = \frac{2}{6} \times \frac{(0,85) + (0,95)}{(0,85) + (1 + 1) + (1) + (1) + (1) + (1)} = 0,09$$

3.3 Similarity of scientific abstracts

Refining the process of semantic comparison of two texts (defined in section 3.2) is performed through a generic structuring of an abstract of a scientific paper into distinct parts whose descriptive roles are different.

Several works have taken interest in the annotation of the discursive structure of scientific papers: text zoning [61] [62]. Their objective is to better characterize the content of the papers by defining several classes (objective, method, results, conclusion, etc.), knowing that the existence of these classes depends on the corpus studied. Categorization is performed at the sentence level. For each sentence of an abstract, authors associate a class chosen from the defined classes.

This work deals with decomposing scientific abstracts into zones for the purpose of detecting plagiarism. From the structure generally reproduced by the authors of scientific papers, the content of a scientific abstract is divided into three distinct parts which are referred to as zones that define the *context*, the *contribution* and the *application domain*. This decomposition is generally reflected in most scientific papers that aim, in principle, at making a scientific contribution in a given domain. This decomposition aims to extract the notions relating to each zone and thus permits a comparison between zones of the same type. The process can then evaluate, in a progressive approach, whether two abstracts deal with the same context, whether their contributions are similar and whether they apply their approach to the same application domain, the risk of plagiarism evidently increasing with each conclusive comparison.

Categorization at the sentence level poses a problem when information from one class is cited in another class. In analyzing several abstracts, it was found that there is no strict uniformity in writing abstracts: all the sentences belonging to a given zone do not contain only the terms describing this zone but may contain terms representing another zone. For example, a sentence assigned to the *application domain* zone may contain terms defining an algorithm or a method (terms that instead define the *contribution* zone). This overlapping of several zones in the same sentence then generates labeling errors.

To illustrate the categorization at the sentence level, each sentence of abstract 2 provided in section (3.3.1), is associated with one of the three selected zones.

"*Recently, new approaches have integrated the use of data mining techniques in the ontology enrichment process.* <context>

Indeed, the two fields, data mining and ontological meta-data are extremely linked: on one hand data mining techniques help in the construction of the semantic Web, and on the other hand the semantic Web assists in the extraction of new knowledge. <context>

Thus, many works use ontologies as a guide for the extraction of rules or patterns, allow to discriminate the data by their semantic value and thus to extract more relevant knowledge. <context>

It turns out, however, that few works aimed at updating the ontology are concerned with data mining techniques. <context>

In this paper, we present an approach to support the onologies management of websites based on the use of Web Usage Mining techniques. <contribution>

The presented approach has been tested and evaluated on an website ontology , which we have constructed and then enriched based on the sequential patterns extracted on the log. <Application domain>"

The following inconsistencies are noted:

- The term *sequential pattern* is assigned to the *Application domain* zone while it represents the algorithm and method used by the author and, therefore, defines the *contribution*.

- The term *Data mining technique* is assigned to the *context* zone while it represents the *contribution*.

- The term *ontologies management* is assigned to the *contribution* zone while it defines the *context*.

To evaluate the semantic similarity of the two abstracts given in section (3.3.1), their content was previously divided as illustrated above. For each abstract, three graphs are constructed and enriched (a graph for each selected zone). For each zone, a similarity value is calculated. The similarity values obtained are very low. This is justified by assigning the terms to a zone while they semantically define another zone, a consequence of the decomposition based on categorization at the sentence level and of the overlapping of zones.

To overcome this problem of overlapping of zones, the terms are assigned to each zone of an abstract according to the overall meaning conveyed by its content. From the global meaning of an abstract, the meaning and the role of its terms are deduced. A term can describe the context of the paper (document categorization, document clustering, image categorization, ontologies enrichment, information retrieval, etc.) or contribution (the methods and algorithms as well as notions used to describe them) or the application domain (classification applied to a given corpus, data mining applied to textual documents, data mining applied to the web, data mining applied to images, etc.). In addition, the terms contained in the title and in the keywords are used, as they often contain information that is not cited in the abstract.

The role of each term is defined according to the knowledge domain in which it is used.

The semantic annotation of the concepts was achieved especially in WordNet Domains [63]. In WordNet Domains, different subject fields are defined, such as medicine, computer science, and architecture. Each synset of WordNet [64] is annotated by one or more Subject Fields where this synset has a meaning. On the basis of the principle that a term describes one of the three zones selected to characterize a scientific abstract, each concept is annotated in the ontology associated with this abstract by one of the three zones (*context*, *contribution* and *application domain*).

The extraction of the concepts corresponding to each zone is performed by projecting the terms composing the content of an abstract on the ontology. The comparison of two abstracts amounts to comparing the zones playing the same role. Three partial similarities are then calculated on the basis of the concepts belonging to the same zone. Two abstracts are compared at three levels. A global similarity of two scientific abstracts *A1* and *A2* is obtained by combining the three partial similarities according to equation (6). The global similarity makes it possible to rank abstracts in descending order of their similarity as illustrated in Tables 10, 11 and 12.

$$\begin{aligned} &Sim(A1, A2) = \\ &\alpha\ sim_{context}(A1, A2) \\ &+ \beta\ sim_{\text{contribution}}(A1, A2) \\ &+ \gamma\ sim_{\text{applicationdomain}}(A1, A2) \end{aligned} \quad (6)$$

α, β, γ are parameters whose values are between 0 and 1. They define the importance attributed to the *context*, the *contribution* and the *application domain*. α + β+ γ=1.

The documents processed are not necessarily suspicious, since it is possible to implement this approach in comparing a document under review, for example, to an entire corpus, without a priori as to its respect for scientific ethics.

A similarity threshold determined by experimentation and according to the ontology and to the collection of abstracts used determines if a risk of plagiarism exists. Abstracts with high similarity will then require a full review of the entire document.

3.3.1 Example

Figure 8 provides an extract of an ontology associated with the domain *ontologies enrichment* and shows the annotation of the concepts by the three zones defined to characterize the content of a scientific abstract.

Let us consider two abstracts from two scientific papers. These papers published in French were translated for the need of our work. The construction of their graphs and calculation of their partial similarities and global similarity is given in section 3.3.2.

Abstract1: *Ontology enrichment based on sequential pattern.*

The mass of information now available via the web, in constant evolution, requires structuring in order to facilitate access and knowledge management. In the context of the Semantic Web, ontologies aim at improving the exploitation of informational resources, positioning themselves as a model of representation. However, the relevance of the information they contain requires regular updating, and in particular the addition of new knowledge. In this paper, we propose an ontologies enrichment approach based on data mining techniques and more specifically on the search for sequential patterns in textual documents.

The presented approach has been tested and evaluated on an ontology of the water domain, which we have enriched from documents extracted from the Web.

Key words: ontology, enrichment, semantic web, data mining, sequential pattern

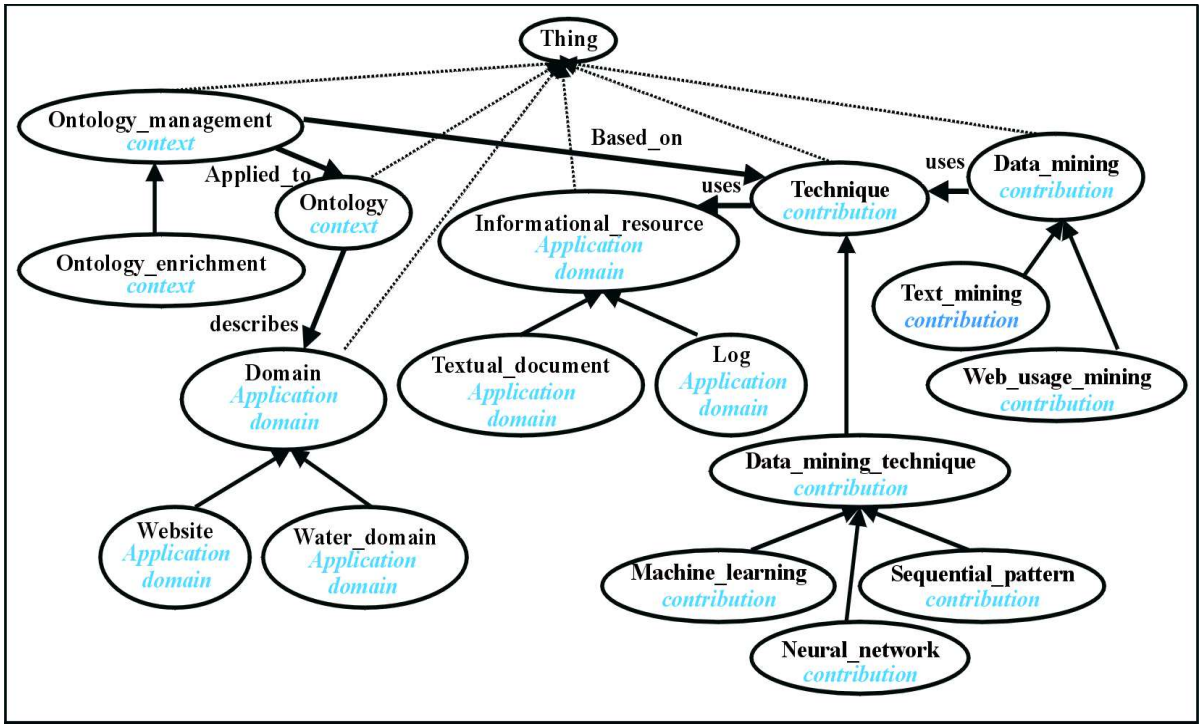

Figure 8: Extract of the *ontologies enrichment* domain ontology, and annotation of concepts by their zone.

Abstract2: *Web usage mining for ontology enrichment.*

Recently, new approaches have integrated the use of data mining techniques in the ontologies enrichment process. Indeed, the two fields, data mining and ontological meta-data are extremely linked: on one hand data mining techniques help in the construction of the semantic Web, and on the other hand the semantic Web assists in the extraction of new knowledge. Thus, many works use ontologies as a guide for the extraction of rules or patterns, allow to discriminate the data by their semantic value and thus to extract more relevant knowledge. It turns out, however, that few works aimed at updating the ontology are concerned with data mining techniques. In this paper, we present an approach to support the onologies management of websites based on the use of Web Usage Mining techniques. The presented approach has been tested and evaluated on an website ontology , which we have constructed and then enriched based on the sequential patterns extracted on the log.

Key words: Semantic Web, ontology, Web Usage Mining, enrichment, data mining, sequential pattern.

3.3.2 Applying our approach

3.3.2.1 Extracting the initial concepts for each abstract

Initial concepts are extracted at the classification step. The two abstracts are attached to the ontology represented in Figure 8. The concepts are assigned to their appropriate zone according to their annotation.

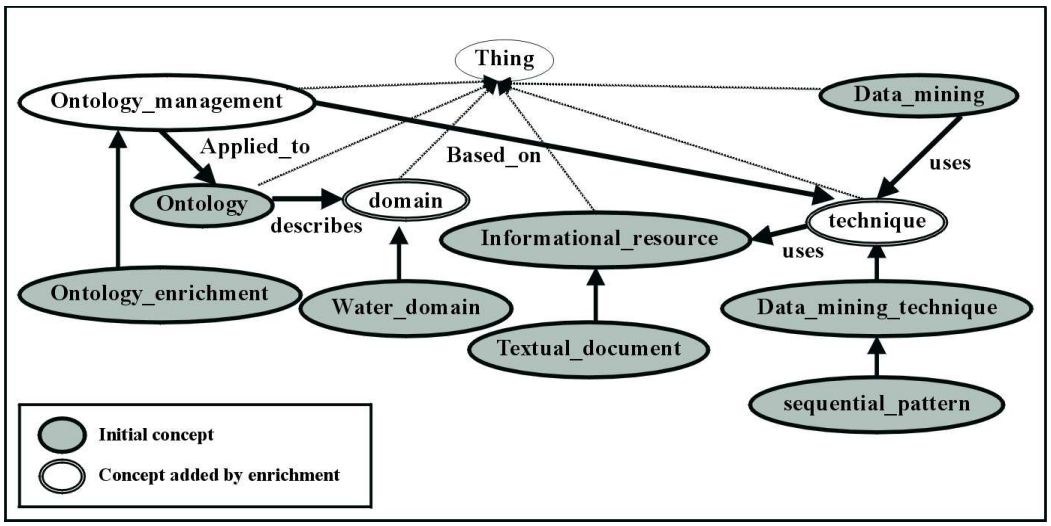

Figure 9: Enriched graph of Abstract1.

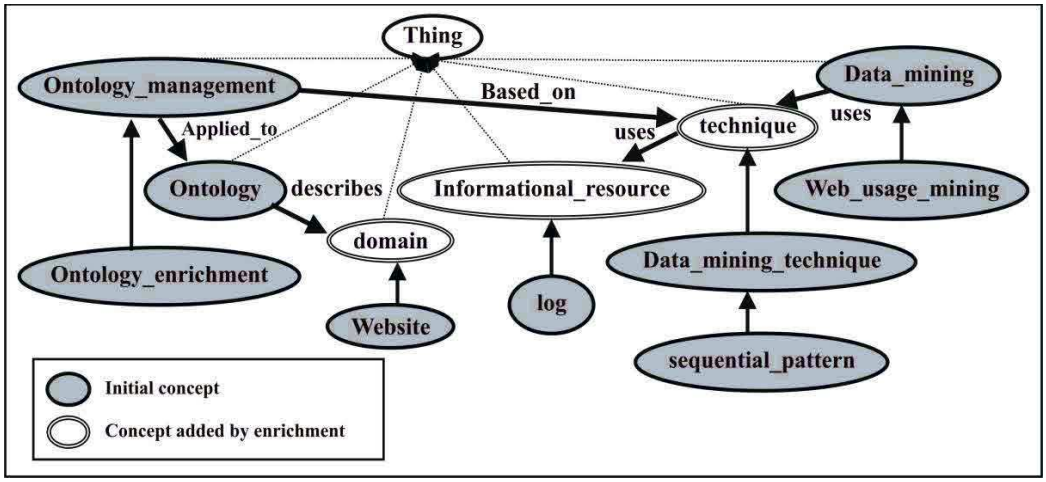

Figure 10: Enriched graph of Abstract2.

	Abstract1		Abstract2	
Zones	**Concepts of Abstract 1**	**Concept type**	**Concepts of Abstract 2**	**Concept type**
context	Ontology_management	Added	Ontology_management	Initial
	Ontology_enrichment	Initial	Ontology_enrichment	Initial
	Ontology	Initial	Ontology	Initial
contribution	Data_mining	Initial	Data_mining	Initial
	Technique	Added	Technique	Added
	Data mining_technique	Initial	Data_mining_technique	Initial
	Sequential_pattern	Initial	Sequential_pattern	Initial
			Web_usage_mining	Initial
Application domain	Informational_resource	Initial	Informational_resource	Added
	Textual_document	Initial	log	Initial
	Domain	Added	Domain	added
	Water_domain	Initial	Website	Initial

Table 5: Distribution by zone of the concepts of Abstract1 and Abstract2.

3.3.2.2 Enrichment of the graphs corresponding to the two abstracts

The *initial concepts* are used to enrich the graphs of the two abstracts by constructing their semantic perimeter and by comparing their graphs. The enriched graphs of the two abstracts *Abstract1* and *Abstract2* are represented in Figure 9 and Figure 10. The distribution by zone of the initial concepts and the added concepts by enrichment is given in Table 5.

3.3.2.3 Similarity calculating between Abstract1 and Abstract2

Table 6 provides values of the global similarity and partial similarities. (Values obtained with α= 0.35, β = 0.63, γ = 0.02, g = 0.05).

$Sim_{context}$(abstract1,abstract2)	0,98
$Sim_{contribution}$ (abstract1,abstract2)	0,59
$Sim_{applicatiodomain}$(abstract1,abstract2)	0,10
Sim (abstract1,abstract2)	0,72

Table 6: Similarities between abstract1 and abstact2.

3.3.2.4 Result

The results obtained indicate that these two abstracts process the same context (sim context = 0.98) with similar approaches. The similarity obtained for the contribution is high (sim contribution = 0.59). These two abstracts differ at the application domain level since the similarity value obtained for this zone is very low (sim application domain = 0.10). The global similarity

obtained is high. This value indicates that the papers associated with these two abstracts should be the subject of a more in-depth analysis that could possibly reveal a case of plagiarism.

4 Experimentations

Our approach is evaluated at two levels. The first evaluation concerns our semantic classification process based on domain ontologies (CBO) and the second concerns the textual similarity calculation process of scientific abstracts.

4.1 Semantic classification process

4.1.1 The data

The implementation of our semantic classification process was performed using WordNet and WordNet Domains simultaneously. In WordNet Domains several knowledge domains are used. These different domains were assimilated to domain ontologies. The Rita similarity measure [13] was used to measure the semantic distance between two synsets in WordNet. The terms within sentences were annotated with their type (noun, verb, adverb and adjective) by Stanford Part-Of-Speech Tagger (POS Tagger) [65].

To evaluate conventional classifiers with our corpus, a pre-processing was performed on the documents. Nouns, verbs and adjectives used in each document were retained. The lemmas relative to these terms were extracted and their weight based on Tf-Idf was then calculated. These lemmas constitute the vector representation of documents. For conventional classifiers, the implementation of three algorithms, SVM, Naive Bayes and decision tree of Weka [66] were used.

Our evaluation covers 10 domains defined in WordNet Domains and a corpus consisting of 976 abstracts of scientific papers. Some abstracts of the domain medicine were extracted from the corpus Muchmore which is a parallel corpus of English-German scientific medical abstracts obtained from the Springer Link web site. All the other abstracts of our corpus were extracted from several scientific journals specialized in the retained domains browsing their Web site. Table 7 gives the distribution of the abstracts relative to the selected domains.

Domains	Number of abstracts
Music	106
Law	83
Computer_science	101
Politics	76
Physics	101
Chemistry	83
Economy	104
Buildings	104
Medicine	117
Mathematics	101
Total	**976**

Table 7: Distribution of abstracts by domains.

4.1.2 Results and discussion

Measures traditionally used in categorization are considered in this work: precision, recall, F-measure and baseline accuracy. The results of our process were compared with those of conventional classifiers. The results obtained are summarized in Table 8.

The recall (*Rc*) determines the number of documents that are correctly classified in a class divided by the total number of documents belonging to that class. Precision (*Pr*) defines the number of documents that are correctly classified in a class divided by the number of documents assigned to that class. A measure that combines precision and recall is their harmonic mean, referred to as the F-measure (*F*). Baseline accuracy (*Acc*) gives the percentage of documents correctly classified relative to the total number of documents in the corpus.

Classes	CBO			Naive Bayes			SVM			Tree C4.5		
	Pr	Rc	F	Pr	Rc	F	Pr	Rc	F	Pr	Rc	F
Music	0,962	0,943	0,952	0,835	0,906	0,869	0,963	0,981	0,972	0,913	0,887	0,900
Law	0,952	0,964	0,958	0,777	0,880	0,825	0,947	0,867	0,906	0,766	0,711	0,737
Computer_science	0,970	0,950	0,960	0,845	0,861	0,853	0,872	0,941	0,905	0,474	0,644	0,546
Politics	0,949	0,974	0,961	0,788	0,829	0,808	0,944	0,882	0,912	0,754	0,645	0,695
Physics	0,960	0,960	0,960	0,833	0,842	0,837	0,887	0,931	0,908	0,513	0,386	0,441
Chemistry	0,940	0,952	0,946	0,947	0,867	0,906	0,986	0,880	0,930	0,848	0,807	0,827
Economy	0,980	0,962	0,971	0,820	0,788	0,804	0,855	0,904	0,879	0,541	0,442	0,487
Buildings	0,980	0,962	0,971	0,950	0,913	0,931	0,925	0,952	0,938	0,757	0,750	0,754
Medicine	1,000	0,983	0,991	0,982	0,940	0,961	0,991	0,991	0,991	0,894	0,863	0,878
Mathematics	0,925	0,980	0,952	0,904	0,842	0,872	0,898	0,871	0,884	0,493	0,673	0,569
Average	**0,964**	**0,963**	**0,963**	**0,872**	**0,869**	**0,870**	**0,926**	**0,924**	**0,924**	**0,694**	**0,682**	**0,683**
Accuracy	**0,963**			**0,869**			**0,924**			**0,682**		

Table 8: Comparison of the results of the various classifiers.

To calculate these different values for SVM, Naive Bayes, and tree C4.5, cross-validation was performed and the results obtained with the best parameters were retained. Table 8 shows that for our process the values of recall and precision are close. These values are close to 1. This is an indicator of the good performance of our classifier. Considering the average of precisions, recalls and F-measure, our process obtains better results than the three conventional classifiers considered. The best percentage of documents correctly classified relatively to all documents in the corpus is obtained by our semantic classification process.

A Wilcoxon Signed-Rank test was used in order to study the statistical significance of the improvement brought about by our process. The p-value between our system and the three conventional classifiers was calculated. This Wilcoxon Signed-Rank test is based on the values of the F-measure obtained for CBO, SVM, Naive Bayes and tree C4.5. This improvement is considered statistically significant if p-value <0.05 and very significant if p-value <0.01. The results of the test are summarized in Table 9.

	CBO - SVM	**CBO - Naive Bayes**	**CBO - Tree C4.5**
P-value (F-measure)	0.00885858	0.00294464	0.000976562

Table 9: Wilcoxon test result.

The p-values obtained with the Wilcoxon test are all less than 0.01. These are very significant p-values. This allows us to conclude that our system significantly improves the classification process of documents compared to conventional classifiers at the threshold $\alpha = 0.01$.

The three conventional classifiers have in common the representation of the documents by words independent of each other as well as a morphological comparison of the words belonging to the documents. The comparison is performed at the word level, whereas in our process, the comparison is performed at the overall context level of the document. A document is represented by the domain described in its content. This domain is deduced by the words of the document taken together considering their relationships in the context in which they appear. In addition, our process is built from domain ontologies, which is a more stable base than a training collection. Indeed, a modification in the choice of the documents constituting this training collection leads to a modification of the results of conventional classifiers.

4.2 Semantic similarity process of scientific abstracts

4.2.1 The data

Our implementation was extended by adding processes to build the semantic perimeters, to divide scientific abstracts into three zones and to compare graphs. To evaluate our approach defining the semantic similarity of scientific abstracts, we constructed an ontology representing the domain of *automatic classification of documents*. To construct our corpus, a set of scientific abstracts related to this domain was extracted from the web. In our different tests, the abstract, the title of the paper and the keywords were taken into account. Each abstract was compared with all the abstracts in the corpus. The abstracts were compared in pairs. For example, the results were obtained by comparing twenty abstracts for which 190 comparisons were made. The construction of the initial graph, the semantic perimeter of each abstract and the comparison of the graphs is done according to the process defined in the previous sections.

Each concept of our ontology was annotated by one of the three selected zones characterizing the content of the scientific abstracts: *context*, *contribution* and *application domain*. This annotation is performed according to the role that each concept plays depending on the chosen domain. For example, *clustering*, *classification* and *document* concepts are annotated by the *context* zone, the concepts representing the different algorithms and methods used by the authors as well as all the concepts describing these methods are annotated by the *contribution* zone. The concepts representing the type of document (*Text, Web*) and the corpus used are annotated by the *application domain* zone.

Our approach was compared to two existing approaches.

The first approach is based on a vector representation of the content of the text: *Bag-of-words*.

The process of extracting terms is similar to the one performed in section 4.1.1. An abstract vector contains the lemmas corresponding to the nouns, verbs and adjectives extracted from the text. Lemmas are represented by their weight based on Tf-Idf. The similarity of two abstracts is calculated by measuring the cosine of the angle between their respective vectors.

The second *n-grams* approach is based on the representation of an abstract by a set of words called *n-grams.* The text is divided into a set of n-grams. The size of an n-gram is determined by a chosen number of consecutive characters, *n*. Several values of *n* were tested (n= 2, 4 and 8) and for each, the similarity between two abstracts was calculated using equation (7) [51] [52] and (8) [53]. For any pair of abstracts *x* and *y*, the similarity Sim(*x,y*) is computed as bellow :

$$Sim(x,y) = \frac{1}{|Dn(x)| + |Dn(y)|} \times \sum_{w \in Dn(x) \cup Dn(y)} \frac{(f_y(w) - f_x(w))^2}{(f_y(w) + f_x(w))^2} \quad (7)$$

$$Sim(x,y) = \frac{|x \cap y|}{|x \cup y|} \quad (8)$$

w denotes an arbitrary n-gram, fx(w) denotes the relative frequency with which w appears in the abstract x

Text1	Text2	Similarity			
		context	contribution	application domain	global
A1.clustering	A3.clustering	1,000	0,401	1,000	0,622
A1.clustering	A10.clustering	1,000	0,295	0,157	0,539
A1.clustering	A2.clustering	0,982	0,306	0,065	0,538
A1.clustering	A9.clustering	1,000	0,227	0,153	0,496
A1.clustering	A16.clustering	1,000	0,169	0,065	0,458
A1.clustering	A15.clustering	1,000	0,103	0,237	0,419
A1.clustering	A17.clustering	1,000	0,092	0,345	0,415
A1.clustering	A5.clustering	1,000	0,095	0,237	0,414
A1.clustering	A18.clustering	1,000	0,022	0,353	0,371
A1.clustering	A19. classif-clust	0,558	0,016	0,065	0,207
A1.clustering	A14.classification	0,244	0,125	0,431	0,172
A1.clustering	A6.classification	0,240	0,074	0,016	0,131
A1.clustering	A8.classification	0,225	0,060	0,541	0,127
A1.clustering	A7.classification	0,225	0,060	0,065	0,118
A1.clustering	A4.classification	0,244	0,036	0,065	0,109
A1.clustering	A11.classification	0,237	0,034	0,108	0,107
A1.clustering	A13.classification	0,230	0,014	0,065	0,090
A1.clustering	A12.classification	0,231	0,007	0,125	0,088
A1.clustering	A20.classification	0,237	0,005	0,031	0,087

Table 10: similarities between A1 and the others abstracts.

Text1	Text2	Similarity			
		context	contribution	Application domain	global
A12.classification	A13.classification	1,000	0,015	0,000	0,360
A12.classification	A4.classification	0,966	0,032	0,000	0,358
A12.classification	A20.classification	0,964	0,012	0,483	0,355
A12.classification	A6.classification	0,965	0,015	0,193	0,351
A12.classification	A14.classification	0,966	0,012	0,066	0,347
A12.classification	A11.classification	0,964	0,007	0,023	0,342
A12.classification	A8.classification	0,900	0,005	0,185	0,322
A12.classification	A7.classification	0,900	0,005	0,000	0,318
A12.classification	A19.classif-clust	0,541	0,107	0,000	0,257
A12.classification	A5.clustering	0,234	0,027	0,329	0,105
A12.classification	A3.clustering	0,234	0,032	0,125	0,105
A12.classification	A18.clustering	0,234	0,026	0,125	0,101
A12.classification	A17.clustering	0,234	0,024	0,123	0,100
A12.classification	A9.clustering	0,227	0,019	0,189	0,095
A12.classification	A15.clustering	0,231	0,004	0,329	0,090
A12.classification	A1.clustering	0,231	0,007	0,125	0,088
A12.classification	A2.clustering	0,233	0,005	0,000	0,085
A12.classification	A16.clustering	0,231	0,006	0,000	0,085
A12.classification	A10.clustering	0,227	0,004	0,032	0,083

Table 11: Similarities between A12 and the others abstracts.

and Dn(x) represents the so called n-gram dictionary of x. | | is the number of n-grams.

The best results were obtained with n = 8 and equation (8), for which the fewest erroneous matching was noted.

4.2.2 Results and discussion

Parameter values (α, β and γ) depend on the ontology and on the corpus used. Several values for these parameters were tested. The goal of this study is to attribute more importance to the *context* zone and the *contribution* zone since it aims to look for matches that primarily indicate documents dealing with the same *context* and similar *contributions*. The following values were retained: α = 0.35, β = 0.63, γ = 0.02, g = 0.05. These values led to the abstracts being grouped based on their context. Table 10 and Table 11 provide the results obtained when comparing respectively the abstracts A1 and A12 with the other abstracts. These tables provide the three partial similarities computed for each pair of abstracts as well as their global similarity. The results, ranked in descending order of global similarity, show a grouping of the abstracts by *context*. Abstract *A1* deals with the *document clustering* context. Abstracts that have the highest similarity with A1 correspond to this context. The abstract *A12* deals with the *document classification* context. Abstracts that have the highest similarity with A12 also correspond to this context hers abstracts.

Table 10 provides a comparison of the similarities between A1 and the other abstracts at three levels. Their similarity can be compared at the *context* level, at the *contribution* level and at the *application domain* level. The values obtained comparing A1 with A3 indicate that these two abstracts deal with the same context (sim context = 1), present similar contributions (Sim contribution = 0, 401) and apply their approach to the same domain (sim application domain = 1). The value of their global similarity is high. These values enable us to retain these two abstracts as suspicious documents, thus requiring further reading and analysis of their entire contents.

Table 11 provides a comparison of the similarities between A12 and the other abstracts at three levels. For the last ten rows of Table 11, very low partial and global similarities were obtained. The first eight rows of Table 11 show that the corresponding abstracts deal with the same context as abstract A12 (sim context >= 0.900) but use different approaches (sim contribution <= 0.032). Their global similarity is low (<= 0,360). This enables us to conclude that abstract A12 does not present any risk of plagiarism with the other abstracts.

The goal of our approach is to be able to find suspicious documents; that is, documents with high similarities. To find these documents, a threshold for the calculated similarity values is determined by experimentation.

To compare the results obtained with our approach to those of *Bag-of-words* and *n-grams*, similarities between the different abstracts of our corpus using the *Bag-of-words* and *n-grams* approaches were calculated. The abstracts were then ranked in descending order of their similarity. For these two approaches, several erroneous matching were found. Table 12, gives an example of the comparison of the similarities between A4 and the other abstracts obtained by our approach, and the *Bag-of-words* and *n-grams* approaches. A4 deals with the context *classification*. With *Bag-of-word* and *n-grams* approaches, most of the abstracts semantically closest to A4 deal with the *clustering* context.

For the *Bag-of-words* approach, abstracts belonging to the context *clustering* (A10, A3, A2, A5, A15, A1) obtain a better similarity score than those (A11, A8, A12, A20, A7, A14) that deal with the same context that A4. It is the same for the *n-grams* approach. Abstracts

Text1	Text2	Our approach	Bag-of-words		N-grams	
A4.classification	A6.classification	0.417272	A06.classification	0.125685	A11.classification	0,042080
A 4.classification	A11.classification	0.401363	**A10.clustering**	0.108323	**A18.clustering**	0,038287
A 4.classification	A13.classification	0.373563	A13.classification	0.097182	**A03.clustering**	0,036313
A 4.classification	A12.classification	0.358287	A19.classif-clust	0.095763	A06.classification	0,035757
A 4.classification	A14.classification	0.358132	**A03.clustering**	0.092988	**A10.clustering**	0,035634
A 4.classification	A7.classification	0.353878	**A02.clustering**	0.092751	A08.classification	0,035602
A 4.classification	A20.classification	0.353120	**A05.clustering**	0.089178	A12.classification	0,034261
A 4.classification	A8.classification	0.330633	**A15.clustering**	0.073636	**A01.clustering**	0,033475
A 4.classification	A19.classif-clust	0.257688	**A01.clustering**	0.066826	A19.classif-clust	0,033400
A 4.classification	A5.clustering	0.191517	**A11.classification**	0.061259	**A07.classification**	0,033071
A 4.classification	A3.clustering	0.180843	**A08.classification**	0.045829	A17.clustering	0,032417
A 4.classification	A9.clustering	0.176679	A18.clustering	0.043951	A09.clustering	0,029097
R4.classification	A2.clustering	0.175801	**A12.classification**	0.042752	A15.clustering	0,026786
R4.classification	A15.clustering	0.147094	A16.clustering	0.041947	A05.clustering	0,025901
A4.classification	A10.clustering	0.135412	**A20.classififcation**	0.033817	**A13.classification**	0,025269
A4.classification	A18.clustering	0.129238	**A07.classification**	0.031982	**A14.classification**	0,023015
A4.classification	A17.clustering	0.119075	A17.clustering	0.028876	A02.clustering	0,020426
A4.classification	A16.clustering	0.114507	**A14.classification**	0.026670	A16.clustering	0,018511
A4.classification	A1.clustering	0.109055	A09.clustering	0.023351	**A20.classification**	0,015968

Table 12: Similarities between A4 and the others abstracts using our approach, Bag-of-words, and N-grams

Abstracts	P5			R-precision		
	Bag-of-words	N-grams	Our approach	Bag-of-words	N-grams	Our approach
A1	1,000	1,000	1,000	0,800	1,000	1,000
A2	0,800	1,000	1,000	0,800	1,000	1,000
A3	0,800	1,000	1,000	0,800	0,900	1,000
A4	0,600	0,400	1,000	0,333	0,556	1,000
A5	0,800	0,600	1,000	0,900	0,800	1,000
A6	1,000	1,000	1,000	0,667	0,778	1,000
A7	0,800	0,800	1,000	0,778	0,667	1,000
A8	0,800	0,800	1,000	0,778	0,556	1,000
A9	0,800	1,000	1,000	0,900	0,900	1,000
A10	0,800	1,000	1,000	0,800	0,900	1,000
A11	1,000	1,000	1,000	0,778	0,889	1,000
A12	0,800	0,800	1,000	0,778	0,667	1,000
A13	0,800	0,800	1,000	0,667	0,667	1,000
A14	1,000	1,000	1,000	0,778	0,667	1,000
A15	0,800	1,000	1,000	0,800	1,000	1,000
A16	1,000	1,000	1,000	0,800	0,900	1,000
A17	0,600	1,000	1,000	0,700	0,900	1,000
A18	0,800	1,000	1,000	0,600	0,800	1,000
A19	1,000	1,000	1,000	1,000	1,000	1,000
A20	0,800	0,800	1,000	0,778	0,667	1,000
Average	**0,840**	**0,900**	**1,000**	**0,762**	**0,811**	**1,000**

Table 13: Precision values for, Bag-of-words, n-grams and our approach.

belonging to the context *clustering* (A18, A3, A10, A1) obtain a better similarity score than those (A7, A13, A14, A20) that deal with the same context as A4.

For all the comparisons made between the abstracts in the corpus, our approach is able to correctly rank the abstracts by context as shown in Table 10, 11, 12 and 13. Clustering and classification are two different contexts. For these two contexts, the methods and algorithms used are different. For that reason, the similarity between two abstracts belonging to these two contexts must be low (low context similarity and low contribution similarity) and, therefore, the risk of plagiarism is very low, or even non-existent. To determine which approach performs the correct matching between abstracts of our corpus, the precision P5 and the R-precision for each approach and for each abstract were computed.

An abstract *Ab1* is assumed relevant to an abstract Ab2, if Ab1 deals with the same context as *Ab2*. Precision Px at point x (x=5, R) is the ratio of the relevant abstracts among the first x returned ones. *R* in the R-precision represents the number of the relevant abstracts to a given abstract in the corpus. Table 13 summarizes the different values.

Our process obtains better results than *Bag-of-words* and *n-grams* approaches. Our process is able to match correctly abstracts dealing with the same context and, therefore, it is more precise than the other approaches.

The Wilcoxon Signed-Rank test was used in order to study the statistical significance of the improvement brought about by our process. The p-value between our system and the two other approaches was calculated.

The results of the Wilcoxon test are summarized in Table 14. The p-values obtained with the Wilcoxon test are all less than 0.01. These are very significant p-values. This leads us to conclude that our system is able to match abstracts by context more correctly than the bag-of-word and n-grams approaches. Others results are summarized in Table 15.

	Our approach / Bag-of-word	Our approach / n-grams
P-value at P5	0.000213431	0.0089409
P-value at R-precision	0.0000638361	0.000219794

Table 14: Wilcoxon test result.

- The content of abstracts *A1*, *A2*, *A3* and *A10* indicates great similarity between abstracts (A1-A3) and (*A2-A10*). These two pairs of abstracts deal with the same context, use the same algorithms and use ontologies to solve similar problematic a priori. As shown in Table 15, our approach makes it possible to select these abstracts as suspicious, while the Bag-of-words and n-grams approaches select only the abstracts (*A1-A3*). *A1* and *A3* use almost the same words in their content. As for the abstracts *A2* and *A10*, their content is described with different words and different sentences, but both are interested in ontology-based feature selection and use the

Text1	Text2	Our approach				Bag-of-Words	N-grams
		context	contribution	Application domain	global		
A1.clustering	A3.clustering	1.000000	0.400673	1.000000	0.622424	0.724688	0,352187
A2.clustering	A10.clustering	0.982456	0.486622	0.112994	0.652692	0.198869	0,050761
A15.clustering	A16.clustering	1.000000	0.188889	0.000000	0.469000	0.470623	0,108580

Table 15: Comparison between Bag-of-words, N-grams and our approach.

same clustering algorithm. Our approach is able to capture the meaning of the abstract and, therefore, retains these two abstracts for a complete examination of their corresponding papers.

- The *Bag-of-words* approach indicates a matching between abstracts *A15* and *A16*. These two abstracts have a high similarity whereas the authors of these two abstracts use different methods in their contribution. Our approach has the advantage of comparing abstracts at three levels. For our approach, the *contribution* similarity between *A15* and *A16* indicates a very low value, which means that the methods used by the authors to solve their problematic are different. This makes it possible to conclude that even if these two abstracts present similar contexts, the risk of plagiarism is low.

Our approach assesses the similarity of texts in two steps. The documents are first assigned to a domain ontology that best describes their content. This overall similarity is achieved by a semantic classification process. This process emphasizes the overall context of the document that can be deduced from the terms of the document taken together, unlike conventional classifiers that consider words independently of each other. For documents belonging to the same ontology, a "local" similarity is calculated. This similarity is based on graphs corresponding to the texts. The enrichment of the graphs through the construction of the semantic perimeter of the texts and comparing of their graphs makes it possible to deduce a similarity not explicitly cited in the texts. The similarity calculation of scientific abstracts is refined by dividing their contents into three zones. Partial similarity values are then calculated. This helps to bring out the notions common to both texts. A grouping by context and a ranking in descending order of the global similarity value can be achieved by combining the three partial similarities. The objective of our approach is to find suspicious documents. It has the advantage of comparing the content of the documents based on three levels. The examination of the similarity obtained for each zone makes it possible to conclude on the existence of a risk of plagiarism.

5 Conclusion

The approach proposed in this paper is meant to assess text similarity. This similarity is based on an overall similarity calculation obtained by a classification process. Our classification process is based on domain ontologies and takes into account the relationships between the terms relative to their context of appearance in the document. The evaluation of our process showed better results than those of conventional classifiers. The construction of the semantic perimeter and the comparison of the graphs of texts based on the domain ontology to which they are attached make it possible to enrich the graphs and to deduce implicit information. Our approach thus present the advantage of taking into account the synonymy and polysemy present in a language and of deducing a similarity between two texts not explicitly cited in their content.

Assessing the similarity between the scientific texts represented by their abstracts is our main interest. In the process of semantic comparison, three distinct parts were defined to structure the abstracts of scientific texts: *context*, *contribution* and *application domain* and three partial similarities were calculated. The comparison of two scientific abstracts is then performed at three levels. The global similarity value of two abstracts, calculated by combining partial similarities, makes it possible to rank the abstracts in descending order of their global similarity. A threshold applied to the calculated similarities is useful in finding suspicious documents and highlighting a risk of plagiarism. Tests were performed on a set of scientific abstracts. The enrichment of the graphs makes it possible to bring out common notions not explicitly cited. Moreover, dividing the contents of abstracts into three distinct zones helps in extracting the notions relative to the *context*, *contribution* and *application domain* and thus makes comparisons between zones of the same type. An evaluation can be made to determine whether two abstracts deal with the same context, whether their contributions are similar and whether they apply their approach to the same application domain.

The quality of our process depends on domain ontologies that must cover the entire vocabulary of the knowledge domain represented for the process to be effective. This may constitute a limitation of this work since the process used does not support the building of domain ontologies. It is, therefore, assumed that they are available. Even if this can be assumed for scientific texts or abstracts structured as shown in this work, the process obviously needs to be refined for it to be used in comparing general texts. Indeed, one of the ways of improving our approach is to generalize the concept of semantic perimeter so as to consider any text rather than just scientific abstracts.

6 References

[1] P. Resnik (1999). Semantic similarity in a taxonomy: An information based measure and its application to problems of ambiguity in natural

language. Journal of Aritificial Intelligence Research, Vol.11, Issue 1, pp. 95-130. https://doi.org/10.1613/jair.514

[2] J. Curran (2002). *Ensemble methods for automatic thesaurus extraction*. In Proceedings of the conference on Empirical methods in natural language processing (EMNLP), Philadelphia, Vol.10, pp. 222-229.

[3] P. Cimano, S. Handschuh, and S. Staab (2004). *Towards the self-annotating web*. In Proceedings of the 13th international conference on World Wide Web, New York, USA, pp. 462-471.

[4] Z.S. Harris (1954). *Distributional structure*. Word, Vol. 10, Issue 2-3, pp. 146–162. https://doi.org/10.1080/00437956.1954.11659520

[5] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman (1990). *Indexing by latent semantic analysis*. Journal of the American Society of Information Science, Vol. 41, Issue 6, pp. 391–407. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9

[6] E. Gabrilovich and S. Markovitch (2007). *Computing semantic relatedness using Wikipedia-based explicit semantic analysis*. In Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, pp. 1606–1611.

[7] M.Yazdani and A. Popescu-Belis (2013). *Computing text semantic relatedness using the contents and links of a hypertext encyclopedia: extended abstract*. In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Beijing, China, pp. 3185–3189.

[8] J. Turian, L. Ratinov and Y. Bengio (2010). *Word representations: a simple and general method for semi-supervised learning*. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, pp. 384–394.

[9] M. Baroni, G. Dinu and G. Kruszewski (2014). *Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors*. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Volume 1, Baltimore, Maryland, pp. 238–247.

[10] C. Leacock, G. A. Miller, and M. Chodorow (1998). *Using corpus statistics and WordNet relations for sense identification*. Journal of Computational Linguistics, Vol 24, Issue 1, pp. 147-165.

[11] R. Rada, H. Mili, E. Bicknell and M. Blettner (1989). *Development and application of a metric on semantic nets.* IEEE Transactions on systems, Man and Cybernetics, Vol 19, Issue 1, pp.17-30. https://doi.org/10.1109/21.24528

[12] Z. Wu and M. Palmer (1994). *Verb semantics and lexical selection*. In Proceedings of the 32nd Annual Meetings of the Associations for Computational Linguistics, Las Cruces, New Mexico, pp. 133-138. https://doi.org/10.3115/981732.981751

[13] D. C. Howe (2009). *RiTa: creativity support for computational literature*. In Proceedings of the seventh ACM conference on Creativity and cognition (C&C '09), Berkeley, California, USA, pp. 205-210.

[14] D. Lin (1998). *An information-theoric definition of similarity*. In Proceedings of the 15th international conference on Machine Learning, pp. 296-304.

[15] P. Resnik (1995). *Using information content to evaluate semantic similarity in a taxonomy*. In Proceedings of the 14th International Joint Conference on Artificial Intelligence, Vol 1, Montreal, Quebec, Canada, pp. 448-453.

[16] S. P. Ponzetto and M. Strube (2007). *Knowledge derived from Wikipedia for computing semantic relatedness*. Journal of Artificial Intelligence Research, Vol 30, Issue 1, pp. 181–212. https://doi.org/10.1613/jair.2308

[17] D. Milne, I. H. Witten (2008). *Learning to link with Wikipedia*. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, California, USA, pp. 509–518.

[18] M. T. Pilehvar and R. Navigli (2015). *From senses to texts: An all-in-one graph-based approach for measuring semantic similarity*. Journal of Artificial Intelligence Vol. 228, pp. 95–128. https://doi.org/10.1016/j.artint.2015.07.005

[19] G. Salton and M.J. McGill (1983). *Introduction to modern information retrieval*. McGraw-Hill computer Science Series.

[20] G. Salton (1971). *The SMART Retrieval System – Experiments in Automatic Document Processing*. Prentice-Hall.

[21] C.J. Crouch, S. Apte, et H. Bapat (2002). *Using the extended vector model for xml retrieval*. In Proceedings of the First Workshop of the Initiative for the Evaluation of XML Retrieval (INEX), Schloss Dagstuhl, pp. 95-98.

[22] E.A. Fox (1983). *Extending the Boolean and Vector Space Models of information retrieval with p-norm queries and multiple concept types*. PhD thesis, Department of Computer Science, Cornell University.

[23] D. Carmel, Y. Maarek, M. Mandelbrod, Y. Mass and A. Soffer (2003). *Searching xml documents via xml fragments*. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, Toronto, Canada, pp. 151– 158. https://doi.org/10.1002/asi.10060

[24] M. Fuller, E. Mackie, R. Sacks-Davis, and R. Wilkinson (1993). *Structural answers for a large structured document collection*. In Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, Pitthsburgh, pp. 204–213.

[25] T. Schileder and H. Meus (2002). *Querying and ranking XML documents*. Journal of the American Society for Information Science and Technology, Vol. 53, Issue 6, pp. 489–503.

[26] T. Joachims (1997). *A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization*. In Proceedings of the Fourteenth

International Conference on Machine Learning, Tennessee, pp.143-151.

[27] S. Jaillet, A. Laurent and M. Teisseire (2006). *Sequential patterns for text categorization*. Journal of Intelligent Data Analysis, IOS Press, Vol.10, issue 3, pp.199–214.

[28] P. Soucy, G. W. Mineau (2001). *A Simple k-NN Algorithm For Text Categorization*. In Proceedings of IEEE International Conference on Data Mining, San Jose, USA, pp.647–648.

[29] A. Hotho, A. Maedche and S. Staab (2002). *Ontology-based Text Document Clustering*. KI, Vol. 16, Issue 4, pp. 48-54.

[30] S. B. Kotsiantis (2007). *Supervised Machine Learning: A Review of Classification Techniques*. Informatica Vol. 31, Issue 3, pp. 249-268.

[31] Y. Yang and X. Liu (1999). *A re-examination of text categorization methods*. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, Berkley, pp. 42–49.

[32] T. Joachims (1998). *Text categorization with support vector machines: learning with many relevant features.* In Proceedings of ECML-98, 10th European Conference on Machine Learning, Chemnitz, Germany, pp. 137–142.

[33] E. Gabrilovich and S. Markovitch (2005). *Feature Generation for Text categorization Using World Knowledge*. In Proceedings of IJCAI 2005: the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, pp. 1048-1053.

[34] A. Hotho, S. Staab and G. Stumme (2003). *Ontologies Improve Text Document Clustering*. In Proceedings of ICDM:3rd IEEE International Conference on Data Minin, Melbourne, FL, USA, pp. 541-544.

[35] H. H. Tar and T.T. Soe.Nyunt (2011). *Ontology-Based Concept Weighting for Text documents*. International Conference on Information Communication and Management IPCSIT vol.16, IACSIT Press, Singapore.

[36] B. Pincemin (2000). *Similarites texte–texts expérience d'une application de diffusion ciblée et propositions*. In Matemáticas y Tratamiento de Corpus, Actes du 2ème séminaire de l'Ecole interlatine de linguistique appliquée,San Millán de la Cogolla, Logroño, Espagne, Logroño : Fundación San Millán de la Cogolla, 2002, pp 35-52.

[37] K. Beyer, J. Goldstein, R. Ramakrishnan and U. Shaft (1999). *When is `nearest neighbor' meaningful*. In Proceedings of ICDT, International Conference on Database Theory, pp. 217-235. https://doi.org/10.1007/3-540-49257-7_15

[38] U.L.D.N. Gunasinghe, W.A.M. De Silva, N.H.N.D. de Silva, A.S. Perera, W.A.D. Sashika and W.D.T.P. Premasiri (2014). *Sentence similarity measuring by vector space model*. In Proceedings of the 14 th International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, Sri Lanka, pp. 185-189.

[39] Y. Liu, C. Sun, L. Lin, Y. Zhao and X. Wang (2015). *Computing Semantic Text Similarity Using Rich Features*. In Proceedings of PACLIC: 29th Pacific Asia Conference on Language, Information and Computation, Shanghai, China, pp. 44 – 52.

[40] J. Lewis, S. Ossowski, J. Hicks, M. Errami and H. R. Garner (2006). *Text similarity: an alternative way to search MEDLINE*. Bioinformatics Vol. 22, Issue 18, pp. 2298–2304. https://doi.org/10.1093/bioinformatics/btl388

[41] E. Yamamoto, M. Kishida, Y. Takenami, Y. Takeda and K. Umemura (2003). *Dynamic programming matching for large scale information retrieval*. In Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages, Vol.11, Sapporo, Japan, pp. 100–108. https://doi.org/10.3115/1118935.1118948

[42] W. Ma and T. Suel (2016). *Structural Sentence Similarity Estimation for Short Texts*. In Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference, Florida, pp. 232–237.

[43] D. Dudognon, G. Hubert and B. Ralalason (2010). *Proxigénéa : Une mesure de similarite conceptuelle*. In Proceedings of the Colloque Veille Strategique Scientifique et Technologique (VSST 2010).

[44] M. Baziz, M. Boughanem, H. Prade and G. Pasi (2005). *A Fuzzy Set Approach to Concept-based Information Retrieval*. In Proceedings of the 4th Conference of the European Society for Fuzzy Logic and Technology and the 11ème Eleventh Rencontres Francophones sur la Logique Floue et ses Applications (Eusflat-LFA 2005 joint Conference), Barcelona, Spain, pp. 1287–1292.

[45] K. M. Shenoy, K.C. Shet, U.D. Acharya (2012). *Semantic plagiarism detection system using ontology mapping*. Advanced Computing: An International Journal (ACIJ), Vol.3, Issue 3, pp. 59–62.

[46] L. Zhang, C. Li, J. Liu and H. Wang (2011). *Graph-Based Text Similarity Measurement by Exploiting Wikipedia as Background Knowledge*. International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol.5, Issue 11, pp. 1328–1333.

[47] W. Jin and R. K. Srihari (2007). *Graph-based Text Representation and Knowledge Discovery*. In Proceedings of the 2007 ACM symposium on Applied computing, Seoul, Korea, pp. 807-811. https://doi.org/10.1145/1244002.1244182

[48] P. Wang, H. Zhang, B. Xu, C. Liu, and H. Hao (2014). *Short Text Feature Enrichment Using Link Analysis on Topic-Keyword Graph*. In Proceedings of Natural Language Processing and Chinese Computing, Springer, pp. 79–90.

[49] J. Leskovec and J. Shawe-Taylor (2005). *Semantic text features from small world graphs*. Workshop on Subspace, Latent Structure and Feature Selection techniques: Statistical and Optimization perspectives, Bohinj.

[50] S. Brin, J. Davis and H. Garcia-Molina (1995). *Copy detection mechanisms for digital documents*. In Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, California, pp. 398–409. https://doi.org/10.1145/223784.223855

[51] C. Basile, D. Benedetto, E. Caglioti, and M. D. Esposti (2008). *An example of mathematical authorship attribution*. Journal of Mathematical Physics, Vol. 49, Issue 12, pp. 125211-1–125211-20. https://doi.org/10.1063/1.2996507

[52] C. Basile, D. Benedetto, E. Caglioti, G. Cristadoro and M. D. Esposti (2009). *A plagiarism detection procedure in three steps: selection, matches and squares*. 3rd Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse, PAN 2009.

[53] B. Stein, S.M. zu Eissen (2005). *Near Similarity Search and Plagiarism Analysis*. In Proceeding of the 29th Annual Conference of the GfKl Springer, pp. 430-437.

[54] R. Lukashenko, V. Graudina and J. Grundspenkis (2007). *Computer-Based Plagiarism Detection Methods and Tools: An Overview*. In Proceeding of the 2007 International Conference on Computer Systems and Technologies - CompSysTech'07, Bulgaria, article N° 40. https://doi.org/10.1145/1330598.1330642

[55] K. Vani, D. Gupta (2015). *Investigating the Impact of Combined Similarity Metrics and POS tagging in Extrinsic Text Plagiarism Detection System*. In Proceeding of the International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, India, pp. 1578-1584.

[56] A. H. Osman, N. Salim, M. S. Binwahlan, H. Hentably and A. M. Ali (2011). *Conceptual similarity and graph-based method for plagiarism detection*. Journal of Theoretical and Applied Information Technology, Vol. 32, Issue 2, pp. 135-145.

[57] D. Rusu, B. Fortuna, M. Grobelnik and D. Mladenić (2009). *Semantic Graphs Derived from Triplets with Application in Document Summarization*. Informatica Vol.33, Issue 3, pp. 357–362.

[58] S. Iltache, C. Comparot, M. Si Mohammed and P. J. Charrel (2016). *Using domain ontologies for classification and semantic interpretation of documents*. In Proceedings of ALLDATA 2016: 2nd International Conference on Big Data, Small Data, Linked Data and Open Data, pp. 76-81.

[59] R. Bendaoud, (2009). *Analyses formelle et relationnelle de concepts pour la construction d'ontologies de domaines à partir de ressources textuelles hétérogènes*. PhD thesis, Henri Poincaré University, Nancy 1.

[60] N. Fuhr and K. Grossjohann (2001). *XIRQL: a query language for information retrieval in XML documents*. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, New Orleans, Louisiana, USA, pp. 172-180.

[61] E. Omodei, Y. Guo, J. P. Cointet and T. Poibeau, (2014). *Analyse discursive automatique du corpus ACL Anthology*. In : Actes de la 21ème conférence Traitement Automatique des Langues Naturelles, Marseille.

[62] Y. Guo, A. Korhonen and T. Poibeau (2011). *A Weakly-supervised Approach to Argumentative Zoning of Scientific Documents.* In Proceedings of the 2011 conference on Empirical Methods in Natural Language Processing, Edinburgh, UK, pp. 273–283.

[63] B. Magnini and G. Cavaglià (2000). *Integrating Subject Field Codes into WordNe*t. In Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation, Athens, Greece, pp. 1413-1418.

[64] C. Fellbaum (1998). *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge MA.

[65] K. Toutanova, D. Klein, C. Manning, and Y. Singer (2003). *Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network*. In Proceedings of HLT-NAACL, pp. 252-259.

[66] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten (2009). *The WEKA Data Mining Software: An Update*. SIGKDD Explorations, Vol. 11, Issue 1. pp. 10-18. https://doi.org/10.1145/1656274.1656278