International Journal of Advanced Research in Engineering and Applied Sciences

ISSN: 2278-6252
Impact Factor: 5.795

ONTOLOGICAL APPROACH APPLIED TO THE C99 ALGORITHM

Rachid Boudouma*

Abstract: *C99 is one of the most famous and popular algorithms in the literature. It was designed for the detection of the topic boundaries in textual documents. C99 like many other algorithms is based on what is called lexical cohesion analysis.*

In this paper, we use this algorithm to locate the theme changes in a specialized text. However, we introduced the elements of domain ontology in order to exploit the semantic linking between terms that appear in such text.

Indeed, we question the pre processing operation used by C99. It performs stemming process by the suffix stripping of the words in order to generate finally the word frequencies matrix; this way of doing generates significant noise which decreases the algorithm performance. We suggest as an alternative to use the ontological analysis process that we have implemented in a previous work.

Finally, we evaluate the effectiveness of this option versus the basic version of C99 by using a specific text corpus formed by concatenated sections dealing with different topics of the domain.

Keywords: *C99, Thematic Segmentation, Domain Ontology, Ontological Approach, topic segmentation.*

*LASTID, Univ of IBN TOFAIL, Kenitra, Morocco

1. INTRODUCTION

For ten years, several works of thematic segmentation were proposed. The majority of them use mathematical and statistical heuristics, especially what is called, in literature, the lexical cohesion study which is based on the analysis of the repetition of words in the text, for example in [Ferret 06], [Hearst 97], [Utiyama 01], [Fernández 07] and [Labadié 09].

This kind of methods assumes that the text segments with a similar vocabulary are likely to be part of a coherent topic segment. It then attaches to find the points where the similarity value presents important variations interpreted as failure of the topic continuity.

Indeed, the robustness of these approaches against the specialized text is not explicitly confirmed. The use of lexicon without considering the semantic linking between terms that appear in the text by several linguistic markers constitutes a handicap for efficient segmentation of this kind of text. Synonymy, hyponymy, meronymy and hyperonymy relations are the most influential phenomena.

In addition, these methods are unable to exploit syntagmatic structures (nominal and verbal) which have a terminological functioning; this represents a shortfall in topic correlation between the various parts of the text.

The introduction of the ontological approach to EnerTex [Fernández 08] in a previous work [Boudouma 13] and [Boudouma 15], has significantly improved the results. In this work we evaluate the impact of ontology introduced on the C99 algorithm [Choi 00] under the same conditions.

C99 is one of the most known algorithms dedicated to topic segmentation of the text. It's built on previous work of Reynar [Reynar 94] and [Reynar 98].

Beforehand, C99 proceed to the elimination of the punctuation and uninformative words from each sentence using a simple regular expression pattern *mateher* and a *stopword* list. A stemming algorithm [Porter 80] is then applied to the remaining tokens to obtain the word *stems*. A dictionary of word *stem* frequencies is constructed for each sentence. This is represented as a vector of frequency counts.

This stemming process based on suffix stripping from words generates significant noise which decreases performance. It made false correlations between text segments through

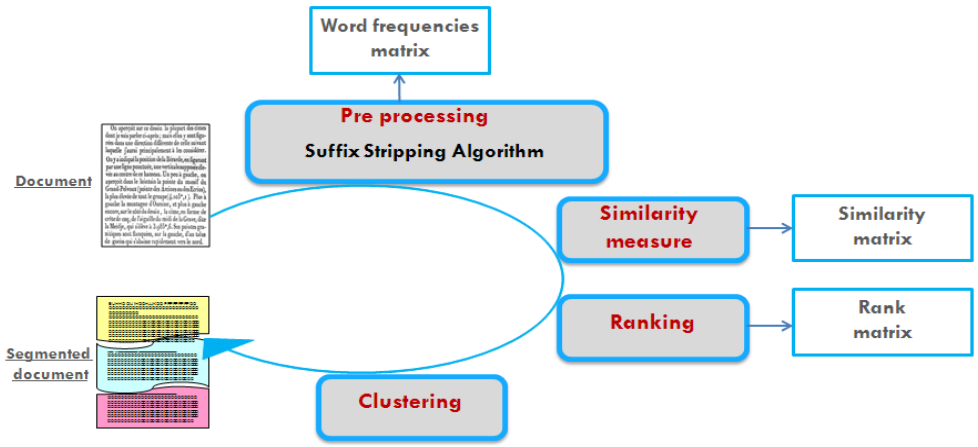
some stems insignificant that are frequent in the text. Furthermore, it often gives some stems morphologically similar with different meanings.

In the remainder of this paper we present firstly an overview on the principle operation and heuristic bases of the algorithm C99, and then we illustrate the improvements we have made to this algorithm.

In the last part we show the results and performance obtained by the thematic detection process *C99* with the new improvements against the scores provided by the basic algorithms according to the assessment protocol.

2. BASIC ALGORITHM C99

The algorithm *C99* performs the segmentation process along three successive steps. The diagram below represents an overview of these steps with their position in the overall algorithm:



Global process of C99 algorithm

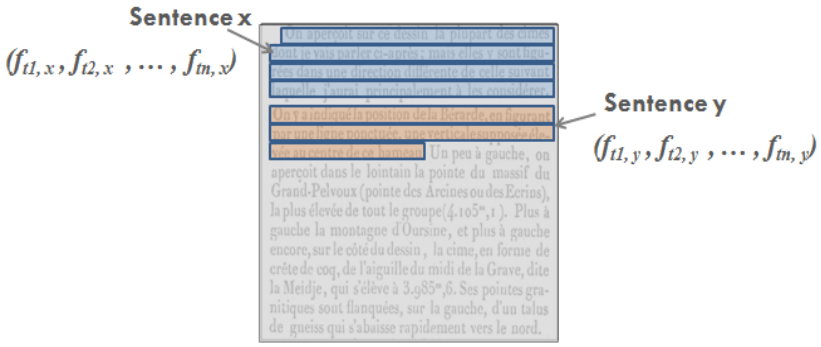
2.1. Pre processing

C99 takes a list of tokenized sentences as input. A tokenizer and a sentence boundary disambiguation algorithm are used to convert a plain text document into the acceptable input format.

Then the punctuation and uninformative words are removed from each sentence using a simple regular expression pattern *mateher* and a *stopword* list. A stemming algorithm [Porter 80] is then applied to the remaining tokens to obtain the word stems.

To stem words, this last algorithm proceeds to suffix stripping of the words. A dictionary of word stem frequencies is constructed for each sentence. This is represented as a vector of frequency counts.

ISSN: 2278-6252
Impact Factor: 5.795



2.2. Similarity measure

The similarity between a pair of sentences ***x,y*** is computed using the cosine measure as shown in the following equation. This is applied to all sentence pairs to generate a similarity matrix.

$$Sim(x,y) = \frac{\sum_j (f_{j,x} \times f_{j,y})}{\sqrt{\sum_j f_{j,x}^2 \times \sum_j f_{j,y}^2}}$$

2.3. Ranking

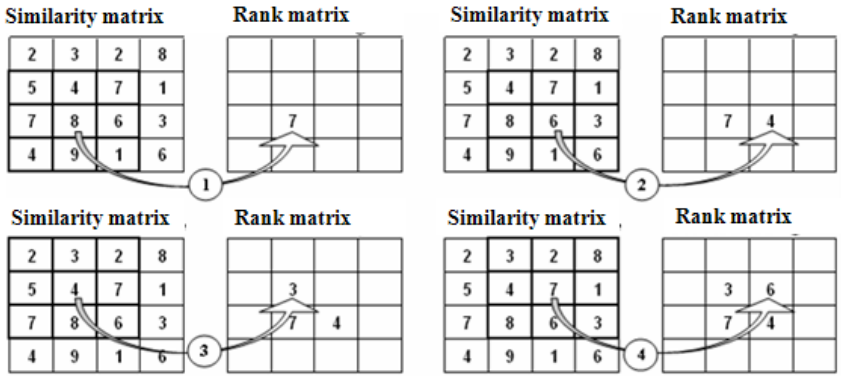
The reliability of this last measure based on the *cosine*, can't be guaranteed when dealing with long lengths of text segments or when these segments have disparities in levels of cohesion (for instance, the introduction section of a document is less cohesive than a section which is about a particular topic).

In this context, it is inappropriate to directly compare the similarity values from different regions of the similarity matrix, that's why C99 propose rather, to compare the ranks of the similarity values.

Indeed, C99 converts the similarity matrix by another called *Rank matrix*. Each value in the similarity matrix is replaced by its rank in the local region. The rank is the number of neighboring elements with a lower similarity value.

$$\frac{nbre\ of\ \ elements\ with\ a\ lower\ value}{nbre\ of\ \ elements\ exa\min ated}$$

The following example shows this procedure:



Example of construction Rank matrix

2.4. Clustering

The final process determines the location of the topic boundaries. The method is based on Reynar's maximisation algorithm ([Reynar 98]; [Helfman 96]; [Church 93]).

A text segment is defined by two sentences ***i, j*** (inclusive). This is represented as a square region along the diagonal of the rank matrix.

Let $s_{i,j}$ denote the sum of the rank values in a segment and $a_{i,j} = (j - i + 1)^2$ be the inside area. $B = \{b_1, \ldots, b_m\}$ is a list of m coherent text segments, s_k and a_k refers to the sum of rank and area of segment k in B.

D is the inside density of B which is written in the form :

$$D = \frac{\sum_{k=1}^{m} s_k}{\sum_{k=1}^{m} a_k}$$

3. ONTOLOGICAL APPROACH

In knowledge engineering, ontology is a notion inspired from philosophy to designate an explicit specification of a conceptualization [Gruber 93]. It is a conceptualization of a domain shared by a community of actors. It is a set of concepts and relationships between them defined by using a formal language understandable by a computer [Roche 06]. Ontology provides domain-specific vocabulary able to represent the knowledge resource content. In addition to these formal and consensual representations, it also provides unique access to knowledge resources through a shared and unambiguous terminology, providing reasoning mechanisms on the modelled knowledge [Bahloul 06].

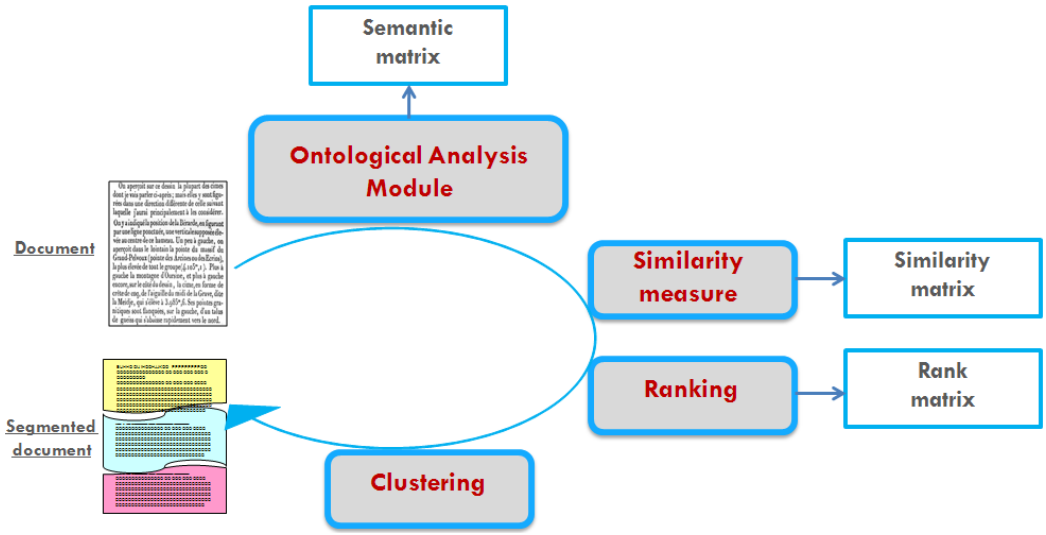
The use of ontological elements (concepts and relationships) in a topic boundaries detection

process provides the following advantages:

- Standardization and normalization which allow treatments and calculations based on a limited number of terminological elements, reducing the size of semantic vectors of the corpus sentences;
- It is quite possible to find two sentences in a text dealing with exactly the same subject and yet having no common term; this problem is largely resolved using ontology which replaces terms by ontological elements that represents them.
- The exploitation of syntagms (nominal and verbal) which have a terminological functioning in corpus; these, in classical approaches, are not represented in the words vectors in document as full words, but by the isolated words that compose them.
- Solving the polysemy problem, that illegitimately creates reconciliations and topic correlation between sentences in the corpus.

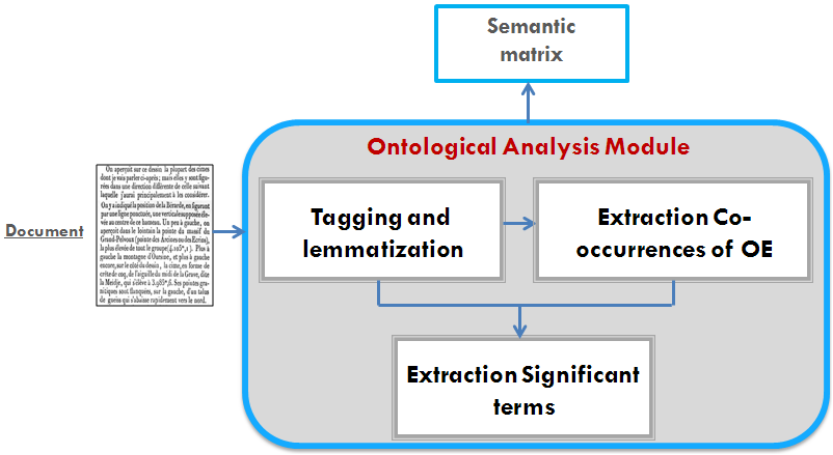
3.1. Improvements to Algorithm C99

To improve C99 we replaced the pre-processing based on the suffix stripping algorithm bay an ontological analysis module. The figure below shows the changes made in the structure of the algorithm:



3.2. Ontological Analysis Module

This module includes three stages as shown by the following scheme:

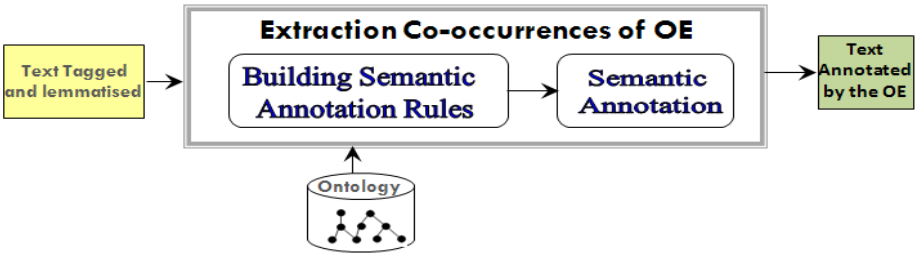


3.2.1 Tagging and Lemmatization

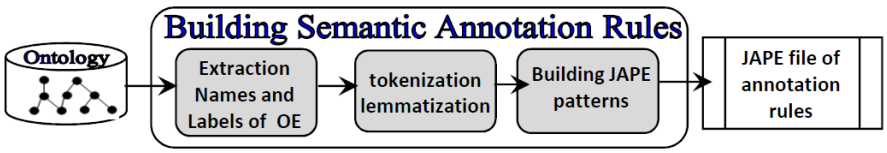
This first stage takes as input a plain text document and uses *TreeTagger* [Schmid 94] as a grammatical tagger of the text for lemmatizing the terms and determining their grammatical categories in order to reduce the number of morphological variances that can be found in the processed text.

3.2.2 Extraction Co-occurrences of Ontological Elements

This part of the algorithm comprises two successive steps. It takes as input the tagged and lemmatised text and the used ontology to give finally as output the annotated text by the Ontological Elements (OE).



1st step: Building Semantic Annotation Rules



It is an algorithm that uses JENA technology to extract from the ontology the useful meta-data (class names, labels, attribute names, relations names, and instance names) for build-

ing automatically the JAPE [Cunningham 00] rules used for annotating ontological elements co-occurrences in the text.

After questioning the ontology to retrieve the list of the ontological elements (OE), the algorithm through this list and for each OE:

1. Get the name of the OE to use as a rule name. This name is used in the header and in the end of the rule ;
2. Built the header and the end of the rule ;
3. For the same OE, gets each '*label*' and starts the construction of the rule body
4. During the construction of the rule body, the words that constitute the label are lemmatised before being used.

Algorithm 1. Building JAPE rules

```
1: OE : Ontological Element
2: OEs : Ontological Elements list
3: rules ← "";
4: For each OE ϵ OEs do {
5:      ruleName ← getNameOf(OE) ;
6:      listLabels ← getLabelsOf(OE)
7:      ruleHeader ← "Rule: "+ ruleName(OE) + "(";
8:      endOfRule ← "):"+ ruleName +"-->:" + ruleName + "." + ruleName + "={kind="" + ruleName +"", rule="+
ruleName + "}";
9:      ruleBody ← "";
10:     For each    label ϵ listLabels    do {
11:          labelBady ← "(";
12:          For each    word ϵ label    do {
13:               wordLem ← lemmeOf(word);
14:               labelBody ← labelBody + "Token.lemma==""+ wordLem +"" ";
15:          }//endFor
16:          if label is not the last {
17:               labelBody ← labelBody +") |"
18:               Else labelBody ← labelBody +")"
19:          }//endif
20:          ruleBody ← ruleBody + labelBody
21:     } //endFor
22:     rules ← rules + ruleHeader + ruleBody + endOfRule;
23: } //endFor
```

24: *//end*

The example below gives an extract of *OFSeT* ontology [Boudouma 13] concerning the concept '*AgentTrain*' expressed in OWL language:

1: *<!--htt p://www:semanticweb:org=ontologies/2010/0/20/OFSeT:owl#AgentTrain-->*

2: *< owl : Class rdf : about = "&ontologies;OFSeT:owl#AgentTrain" >*

3: *< rdfs : label xml : lang = "fr" >* agents de trains *< /rdfs : label >*

4: *< rdfs : label xml : lang = "fr" >* agents des trains *< /rdfs : label >*

5: *< rdfs : label xml : lang = "fr" >* agents du service des trains *< /rdfs : label >*

6: *< rdfs : label xml : lang = "fr" >* brigade de conduite *< /rdfs : label >*

7: *< rdfs : label xml : lang = "fr" >* personnel de conduite *< /rdfs : label >*

8: *< rdfs : label xml : lang = "fr" >* personnel des trains *< /rdfs : label >*

9: *< rdfs : label xml : lang = "fr" >* personnel train *< /rdfs : label >*

10: *< rdfs : subClassOf rdf : resource = "&ontologies;OFSeT:owl#Agent"*

11: */ >*

The JAPE rule produced for this example by the algorithm 1 will have the following form:

Algorithm 2. JAPE rule 'AgentTrain'

1: *Rule : AgentTrain*

2: *(*

3: *(Token:lemma == "*agent*" Token:lemma == "*de*" Token:lemma == "*train*")*

4: *| (Token:lemma == "*agent*" Token:lemma == "*du*" Token:lemma == "*train*")*

5: *| (Token:lemma == "*agent*" Token:lemma == "*du*" Token:lemma == "*service*" Token:lemma == "*du*" Token:lemma == "*train*")*

6: *| (Token:lemma == "*brigade*" Token:lemma == "*de*" Token:lemma == "*conduite*")*

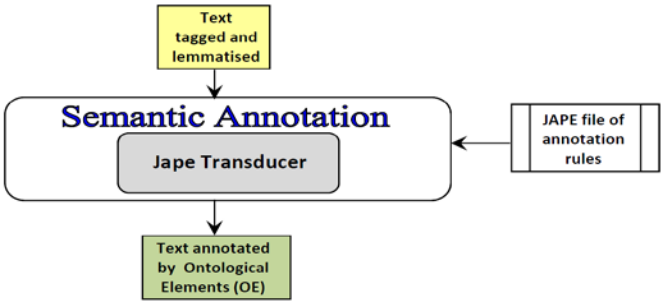
7: *| (Token:lemma == "*personnel*" Token:lemma == "*de*" Token:lemma == "*conduite*")*

8: *| (Token:lemma == "*personnel*" Token:lemma == "*du*" Token:lemma == "*train*")*

9: *| (Token:lemma == "*personnel*" Token:lemma == "*train*")*

10: *) : AgentTrain-- >: AgentTrain:AgentTrain = kind = "AgentTrain"; rule = AgentTrain*

2nd step: Semantic Annotation

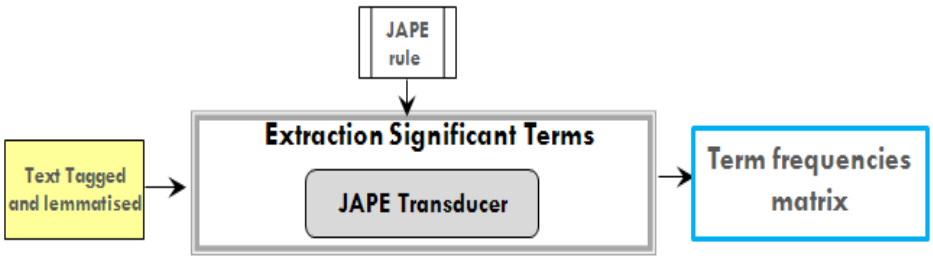


This module is responsible of the research and the semantic referencing of the various linguistic forms that are matched by the rules built in the previous phase, using "JAPE transducer" proposed in GATE. We get in the output, a text annotated by various ontological elements.

3.2.3 Extraction Significant Terms

This step allows extracting the significant terms through a JAPE rule. The relevant terms that we have chosen are those of grammatical categories **nouns** and **verbs**.

Our motivation to select only these categories comes from the fact that the nouns and verbs are the key elements for building the speech meaning. Of course, the construction of the exact meaning is more complex process than that.



We have technically applied this heuristic using transducers GATE [Cunningham 02] by a JAPE rule that we have implemented and named *SignifTerm*.

Algorithm 1. JAPE rule 'SignifTerm'

```
1: Rule : SignifTerm
2:  (
3:      ({Token.category=~"VER:", Token.lemma !="être", Token.lemma !="avoir", Token.lemma !="exister",
Token.lemma !="faloir", Token.lemma !="devoir", Token.lemma !="pouvoir", Token.lemma !="faire", To-
ken.lemma !="agir" , Token.string !="l", Token.string !="d", Token.string !="s", Token.string !="qu"})

4:      | (({Token.category=="NOM", Token.lemma !="cas", Token.lemma !="exemple", Token.string !="l", To-
ken.string !="d", Token.string !="s", Token.string !="qu", Token.string !="jusqu"})
5:      | ({Token.category=="NAM"})
6:     ) : SignifTerm -- >: SignifTerm: SignifTerm = kind = " SignifTerm"; rule = SignifTerm
```

The *SignifTerm* rule excludes some non-significant terms such as (être, avoir, exister, devoir, falloir, cas, ...). The list of those eliminated words is not limited, it can be extends by others.

3.3. Semantic vectors : Adapted Salton's model

This last module builds in fine the semantic vector for each sentence using *Salton*'s model with a slight adaptation:

Let *O=(C,R)* is a domain ontology where:

$$C = \{c_1;\ c_2;\ \ldots;\ c_i;\ \ldots;\ c_n\}$$

and

$$R = \{r_1;\ r_2;\ \ldots;\ r_i;\ \ldots;\ r_m\}$$

C and *R* are respectively, the sets of concepts and relationships of the ontology *O*.

Let a textual document composed of *k* sentences and *C'* is its ontological elements set (having co-occurrences in the text). We can deduce that: $C' \subset \{C ; R\}$.

Let *St* is the significant term set extracted from the document. We define the basis *B*={ *C' ; St*} with cardinal *N.*

The *k* sentences are represented in *B* basis by the semantic vectors as follows:

$$\begin{aligned} X^1 &= (x_1^1, x_2^1, \ldots, x_j^1, \ldots, x_N^1) \\ X^2 &= (x_1^2, x_2^2, \ldots, x_j^2, \ldots, x_N^2) \\ &\vdots \\ X^i &= (x_1^i, x_2^i, \ldots, x_j^i, \ldots, x_N^i) \\ &\vdots \\ X^k &= (x_1^k, x_2^k, \ldots, x_j^k, \ldots, x_N^k) \end{aligned}$$

Where x^i_j is the frequency of the element E_j of the basis *B* in the sentence *i*

4. EXPERIENCES AND RESULTS

4.1. Experience Protocol

So that we can evaluate our improvements to *C99* algorithm, we used a test text that we have already used in a previous work [Boudouma 13] and [Boudouma 15].

This corpus is constituted by concatenated paragraphs dealing with different topics of the railway safety domain; it has been prepared by the railway domain experts who have specified manually the thematic borders.

Table 1. ***TEXT CARACTÉRISTICS***

Words Number	6500
Sentences Number	150
Themes Number	30
Nbr words/Sentence	44
Nbr Sentences/Theme	5
Thematic borders	3, 6, 9, 13, 18, 26, 31, 37, 42, 45, 49, 52, 56, 58, 61, 65, 67, 72, 79, 84, 91, 99, 106, 115, 120, 124, 134, 142, 147

The C99 script used in the evaluation is the one published in the website: *http://picard.at.northwestern.edu/morphadorner/documentation/javadoc/edu/northwestern/at/morphadorner/corpuslinguistics/textsegmenter/c99/C99.html*

The evaluation focuses on the comparison of the results obtained under optimum conditions, by the two versions of the algorithm *C99* (basic version and Improved one). These results are expressed as conventional indicators (*recall, precision, F-score* and *WindowDiff*)

4.2. Evaluation Results

The evaluation will focus on a comparison performed on the same test corpus and with the optimal parameters of each version of the algorithm.

On *C99* and its improved version, we obtained results by varying the parameter (*ranking mask*). The averages of experimental results are summarized in the table below:

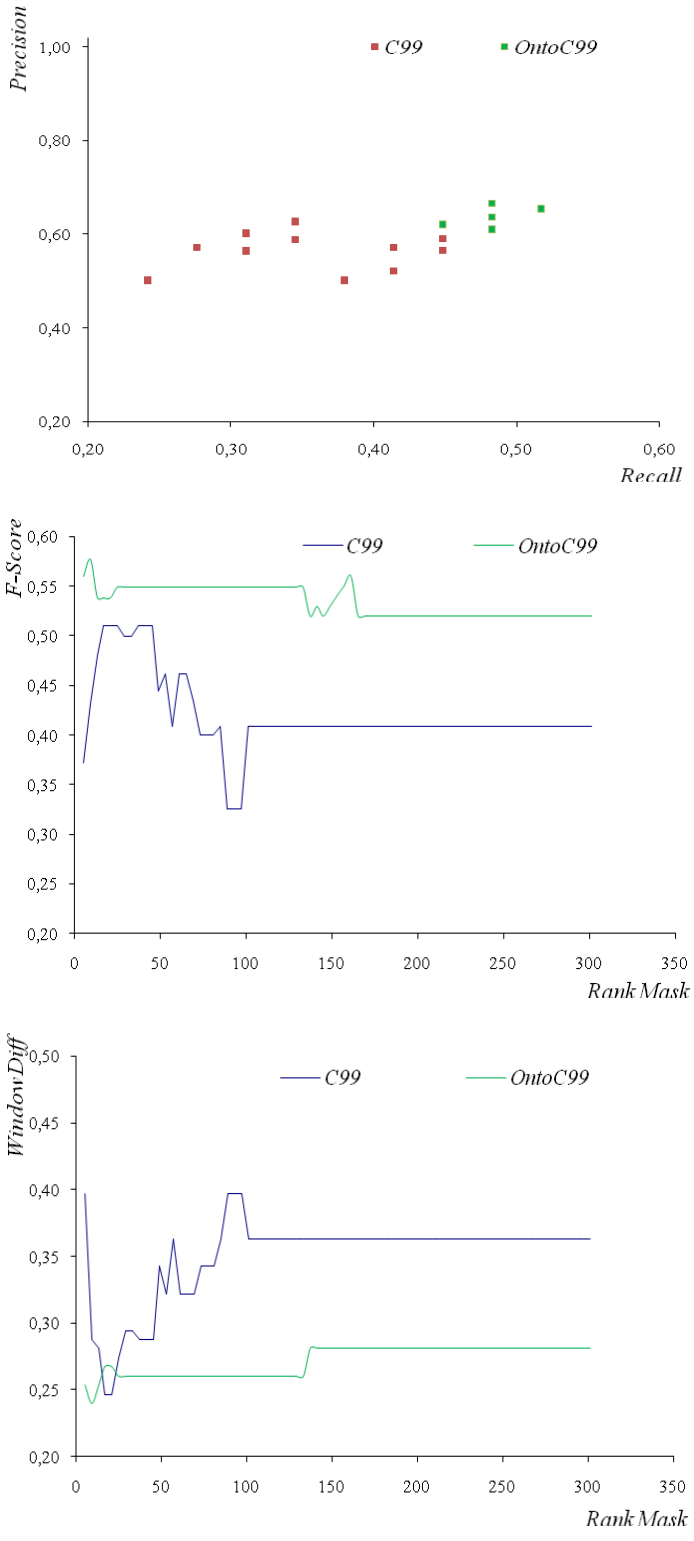
Table 2. ***RESULTS OF EXPERIENCE***

	C99	***Improved C99***
Recall	0,33	0,46
Precision	0,59	0,63
F-score	0,42	0,53
WD	0,35	0,27

We observe that improved *C99* gives the best results for all markers used (46% *recall*, 63% *precision*) which is distant from basic version (33% *recall*, 59% *precision*). The same finding was recorded in terms of *windowDiff* with a rate of 0.27 against 0.35.

The comparisons of all results are graphically visualized in the following figures; note that

OntoC99 is the improved algorithm:



5. CONCLUSION

In this work we presented ontological approach applied à C99 algorithm to detect thematic boundaries in a specific text. So by exploiting the domain ontology, we offer an alternative

against using the lexicon. Thus, we used that rather than the pre processing operation used by C99 based on the suffix stripping of the words.

We have diagrammed the changes and improvements brought to *C99*; however, a summary of our ontological approach has been showed as it has been raised in our previous work. In fact, we presented the operating mechanisms of its various modules as well as its basic technologies and heuristics.

The evaluation of our improvements, on a test corpus of railway domain, has shown interesting results against the basic version of the algorithm.

REFERENCES

1. [Bahloul 06] Bahloul Djida Une approche hybride de gestion des connaissances basée sur les ontologies : application aux incidents informatiques z. in thesis pp 128, 2006.
2. [Boudouma 13] Boudouma Rachid, Raja Touahni et Rochdi Messoussi (2013). New approach for topic segmentation of railway text. In ZENITH: INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH Vol. 3 Issue 2, Feb 2013 issue of ZIJMR HINDIA.
3. [Boudouma 15] Rachid Boudouma. *SeThemO* : Thematic segmentation-based ontology. In IJARIE : INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN IT AND ENGINEERING Vol. 4 Issue 3, March 2015.
4. [Choi 00] Fred Y. Y. Choi. n Advances in domain independent linear text segmentation. z. Proceeding of NAACL-00, pp 26–33, 2000.
5. [Church 93] Kenneth W. Church. 1993. Charalign: A program for aligning parallel texts at the character level. In Proceedings of the 31st Annual Meeting of the ACL.
6. [Cunningham 00] Cunningham. H, D. Maynard and V. Tablan (2000). JAPE: A Java Annotation Patterns Engine, Department of Computer Science, University of Sheffield, 2000.
7. [Cunningham 02] Cunningham H, Maynard D, Bontcheva K, Tablan V (2002). GATE : A framework and graphical development environment for robust NLP tools and applications. In Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia, US.
8. [Ferret 06] Olivier Ferret. n Approche endogène et exogène pour améliorer la seg-

mentation thématique de documents z. TAL, 2006.

9. [Fernández 07] Énergie textuelle de mémoires associatives. TALN 2007, Toulouse, 5–8 juin 2007
10. [Fernández 08] Fernández S., Sanjuan E. & Torres-Moreno J. M. (2008). Enertex : un système basé sur l'énergie textuelle. In TALN 2008, Avignon, 9-13 juin 2008.
11. [Gruber 93] Gruber. T.-R (1993). Translation Approach to portable Ontology Specifications. In : Knowledge Acquisition, 1993, vol.5, N°2, pp199-220.
12. [Hearst 97] M. A. Hearst. n TextTiling : Segmenting text into multiparagraph subtopic passages. z. Computational Linguistics, pp 33–64, 1997.
13. [Helfman 96] Jonathan I. Helfman. 1996. Dotplot patterns: A literal look at pattern languages. Theory and Practice of Object Systems, 2(1):31-41.
14. [Labadié 09] Alexandre Labadié Segmentation thématique de texte linéaire et non-supervisée : Détection active et passive des frontières thématiques en Français z. Dans thèse pp 5-25, 2009.
15. [Porter 80] M. Porter. 1980. An algorithm for suffix stripping. Program, 14(3):130-137, July.
16. [Roche 06] Roche Christophe (2006): Colloque 2006 de la Société française de terminologie « Terminologie et ontologie : descriptions du réel » Ecole Normale Supérieur – 1er Décembre 2006)
17. [Reynar 94] Jeffrey C. Reynar. 1994. An automatic method of finding topic boundaries. In Proceedings o/ ACL'9,~ (Student session).
18. [Reynar 98] Jeffrey C. Reynar. 1998. Topic segmentation: Algorithms and applications. Ph.D. thesis, Computer and Information Science, University of Pennsylvania.
19. [Schmid 94] Helmud Schmid. Probabilistic part-of-speech tagging using decision trees. In Proceedings of the International Conference on New Methods in Language Processing, p. 44–49.
20. [Utiyama 01] M. Utiyama et H Isahara. n A statistical model for domain independent text segmentation z. ACL, pp 491–498, 2001.