Bilingual Information Retrieval using a Parallel Platform

Alberto Márquez, Darnes Vilariño, Erick Pinacho,
Mireya Tovar, and Beatriz Beltrán

Abstract. This paper discusses the problem of Bilingual Information Retrieval over bilingual documents using lexical resources and implementation over a parallel Kraken Platform designed by [1]. The system uses bilingual documents (English and Spanish), which are pre processed and post processed to group these documents using metric as the title, proper names and the first paragraph of the document. It was subsequently used the platform to send clusters obtained and the query to each slave lifted in the Platform Kraken and each slave showing the similarity to the query.

Keywords: Clustering algorithms, unsupervised learning, parallel system, incremental algorithms.

1 Introduction

In the last few years, natural language processing (NLP) techniques and tools have been incorporated into information retrieval (IR) systems with varying degrees of success, it has taken interesting among researchers, when they have to devise new techniques to assist user in the process of searching. Currently one of the richest in containers information is the Web, but its growth has generated problems at the time of information retrieval, mainly due to repetitive information, diversity of data, lack of standardization to represent data, and others. This become even more complex if they are to a query as provided in both English and Spanish will return documents relevant to it.

Since the 1980s, corpus linguistics has developed at an accelerated speed. While the construction and exploitation of English language corpora still dominate the research of corpus linguistics, corpora of other languages, particularly typologically related European languages such as French, German and Portuguese and Asian languages such as Chinese, Korean and Japanese, have also become available and have notably added to the diversity of corpus-based language studies. In addition to monolingual corpora, parallel and comparable corpora have been a key focus of non-English corpus linguistics, largely because corpora of these two types are important resources for translation and contrastive studies.

A parallel corpus can be defined as a corpus that contains source texts and their translations. Parallel corpora can be bilingual or multilingual. For a comparable corpus, the sampling frame is essential. The components representing the languages involved must match with each other in terms of proportion, genre, domain and sampling period.

This paper presents a parallel system to process bilingual text (English and Spanish) for cross language querying, the sets of metrics are dependent on lexical resources, and linguistic tools. Which cluster is send to Slave Node with query, which

Advances in Computer Science and Artificial Intelligence.
Research in Computing Science 39, 2008, pp. 199-209

Received 23/06/08
Accepted 01/08/08
Final version 10/09/08

Slave Node returns the comparison between the clusters of documents and query. It is verified in a practical way that a trading system of machine translation can employ approximately ten months to provide the translation of 250 000 documents, this is not feasible for a real system of IR. In this system only used a first paragraph of document.

2 Metrics

Several systems make use of independent modules, each assigned a score (weight) to each unit (words, sentences, paragraphs, etc.). Subsequently a module in particular is responsible for making the combination of different weights assigned to each unit getting a single value for each unit; eventually the system will return n the first units with the highest weight. Below are the most commonly used techniques.

a) **Proper Names**: It is a kind of attribute that refers to individuals and/or locations indicate that the sentence could contain important information. One example is the news of a natural disaster, the sentence containing the name of the place where the damage occurred could be important, which is a point of comparison with other documents. This paper takes its own name as one that begins with capital letters, which is not an empty word or closed and it is not possible to find its [12].

b) **Similar to title**: Several systems make use of keywords to identify relevant sentences [6]. But as we know it is impossible to define a set of keywords that apply to all kinds of documents because doing so would force the system to depend on the thematic domain and further language. With what is considered is with the title. Often the title of a document of great information on the contents. This is why it was decided to use this attribute. When there is no title, the first sentence of the document is taken as the title and thus prayers with greater similarity to the title are considered important [13] - [16].

c) **First paragraph**: It has been shown that the terms are both in the title as in the first paragraph largely reflect the theme of a document, these results have been exhibited in [17].

To compare two sentences are taken on a role of similarity, this is done by taking a value greater than a threshold defined β (between 0 and 1). The function $Sim\ (O_i, O_j) \geq \beta$ [18] to see (1).

$$sim(O_i, O_j) = \frac{|O_i \cap O_j|}{|O_i \cup O_j|} \quad (1)$$

Where Oi and Oj are a set of words. Another method is to use the Point of Transition (PT), which separates the words of a document in terms of high and low frequencies. The words about the point of transition are the most relevant. This method is ideas based on investigations by the formula for calculating the PT is presented in (2).

$$PT = \frac{\sqrt{1 + 8 \times I_1} - 1}{2} \quad (2)$$

Where I_1 represents the number of words frequently 1. According to the characterization of frequencies mean the PT can occur in the vocabulary of a text by identifying the lowest frequency of high, which is not repeated. For example, with the words around the PT by 25%, it generates a virtual paragraph. The results when applying the metrics are favorable in information retrieval.

3 Clustering Algorithms

Clustering algorithms heuristically build clusters from sets of objects which are characterized by several features and a similarity function. These algorithms try to maximize the similarity between objects in the same cluster and/or minimize the similarity between objects in different cluster. Clustering algorithms are used in many fields of non formalized sciences like information retrieval, data mining genetics, computer vision, biology, earth science, and others.

Essential elementals for to resolve the problem of clustering are: the representation of space objects, the measure of similarity between objects (not necessarily a distance) and clustering or heuristic approach to implement. In some of these methods is necessary to define also a measure of similarity between the clusters, which is defined in terms of similarities between objects that make up clusters [10].

We used the single-pass algorithm [11]. There are a small number of clustering algorithms which only require one pass of the file of object descriptions. Basically they operate as follows:

1. – The object descriptions are processed serially.
2. – The first object becomes the cluster representative of the first cluster.
3. – Each subsequent object is matched against all cluster representatives existing at its processing time.
4. – A given object is assigned to one cluster (or more if overlap is allowed) according to some condition on the matching function.
5. – When an object is assigned to a cluster the representative for that cluster is recomputed.
6. – If an object fails a certain test it becomes the cluster representative of a new cluster.

Once again the final classification is dependent on input parameters which can only be determined empirically (and which are likely to be different for different sets of objects) and must be specified in advance. For the clusters of documents were used metric described above.

4 Kraken Plataform

The search for solutions to real problems usually requires using a lot of calculations. This causes delivering results is too late. One solution to this problem is offered by parallelism, which allows several operations at once, favoring reducing the time required to execute a task.

Java provides mechanisms for competition, management functions and low-level technologies for developing distributed applications. These and other features that owns Java, they do an excellent candidate to develop software side, for details see [18]. The platform contains elements that are listed below:

a) **Node's Name**: The platform is shaped by a set of computers, which will be running the Demon Slave, Master Demon, or both. When you start the Demon in a particular computer, the Demon is responsible for announcing to the rest of the Platform that the node is added. The Platform is responsible for distributing classes between the nodes. For sending messages is necessary to know the addressee, and sometimes the source.

b) **Sokets**: To make communication across the platform is used in a socket UDP multicast. The use of UDP socket implies that the package delivery is not guaranteed, nor the order. However, the platform is targeting a cluster or a computer network LAN. IP addresses are among 244.0.0.0 and 239. Even used to multicast. Regardless of the direction given to any network interface can be used any direction in the range previously indicated in any application. The address in the range indicated, plus a number of standards UDP port, and can receive and send messages on multicast.

c) **Shared Memory**: The platform offers a building which serves as shared memory. With it, you may give the developer a region of memory that is shared among all the nodes of the *Platform*. The region of memory offered is seen by all the nodes of the platform, using replicas. Each node has a complete replica of shared memory. The Shared Memory is designed to perform a disco paging through pages of small size. With this, it gives the developer a memory size theoretically limited by the capacity of disk.

d) **Synchronization:** Several hardware platforms offer parallel mechanisms synchronization. Depending on the type of platform, processors can be synchronized by the control unit-level instruction, or you can synchronize at regular intervals at certain points of execution. The synchronization points are a common construction in the software tools for developing parallel and distributed applications. At software, a point of synchronization is to ensure that the different running processes reach a point in the execution in which processes one-one will stop until they all arrive there, when his execution continued.

All functionality accessible user classes are grouped within the class *PlatformBind*. One such instance is communicated to each user's task that is created within the Platform, and this is done through the interface *ClassStub*. All user tasks to be run in

the Platform must implement the interface *ClassStub*. With this class of user must implement methods setProperties() and *execute ()*.

Through the method *setProperties()* will have access to the user's task an instance of *PlatformBind* who used to use the features of the Platform. This method will be invoked before the implementation of the method *execute()*. The method *execute()* must meet the code that the user wishes to be executed. With the implementation of the interface *ClassStub*, will be achieved expose the functionality of the user platform.

5 System Design

This section presents the design for the construction of information retrieval systems in Parallel (RIP). The objective, as mentioned before, is to develop a system that given a query by the user, it can retrieve documents in both English and Spanish. The most important components for the development of the system are illustrated in the figure below:

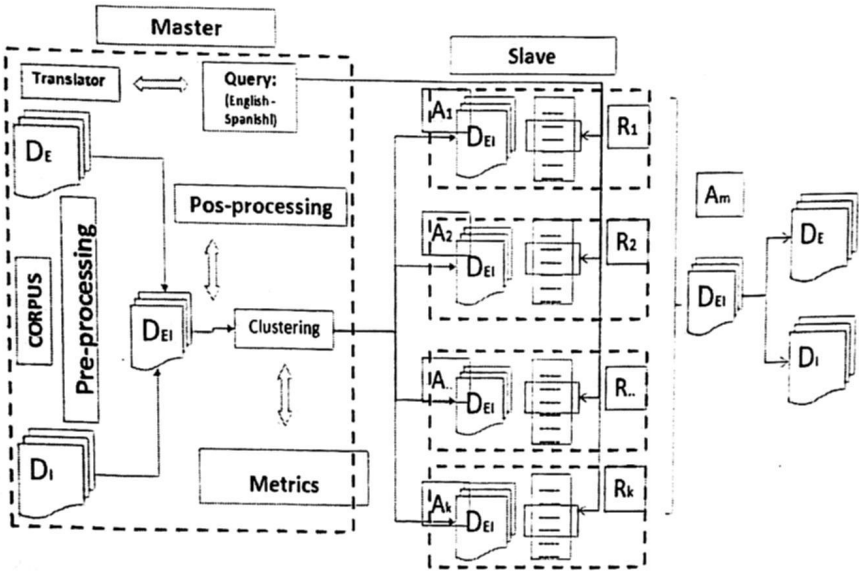

Fig.1. RIP System Scheme.

· Description:

a) **Corpus**: Our corpus procedures were initially developed to process D_I (English documents) and D_E (Spanish documents), but are extendible to other languages.

b) **Pre processing**: For each document (English - Spanish), it was divided to sentences, we get the titles, we get proper name, we get the language and to extract the text that will be translated.

c) **Post processing**: It removes stopwords (both languages), titles, proper name and text to translate, and to stemmer.
d) **Clustering** ($A_1..A_k$): The grouping of documents according to the characteristics obtained (proper names, paragraph original-translate, titles original-translate) by an algorithm *single pass*.
e) **Query**: We get the user's query and this is processed, to translate, to remove stopwords and to stemmer.
f) $R_{1..n}$: It is the representation of each set of documents previously grouped. The representation is divided into two parts (English-Spanish) each obtained through implementing Paragraph Virtual Point Transitional

g) D_{EI}: It is the representation of each set of documents previously grouped.

The communication scheme proposed for the parallel algorithm is the master-slave, and then defines the tasks which will make the master and slave respectively.

a) Master
   - ♣ Pre processing of corpus.
   - ♣ Post processing of corpus.
   - ♣ Generating groups (*k*).
   - ♣ Send *k-groups/n-tasks*.
   - ♣ Send query to each Slave.

b) Slave
   - ♣ Receive *k* groups.
   - ♣ Receive query.
   - ♣ 2 process (English - Spanish).
   - ♣ Each process:
     - Convert *t* documents to 1 document.
     - Generate Point Transition (PT).
     - Make the Virtual Paragraph (PV).
     - Evaluate query with PV.
     - Show the similarity.

6 System Implementation

The search for solutions to real problems usually requires using a lot of calculations. This causes delivering results is too late. One solution to this problem is offered by parallelism, which allows several operations at once, favoring reducing the time required to execute a task. The overall activities being carried out on the platform are:

a. **Starting the Master Daemon**: It is controls the platform.
b. **Starting the Slave Daemon**: It lifted the demons slaves for each node.
c. **Registration Demons Slaves**: Each slave demon informs to slave master and its existence is saved by the master slave.

d. **Receiving Task Definition**: According to the definition of user classes, the master demon communicates with the demons slaves to the distribution of classes.
e. **Execution of tasks:** Each node where there is a demon slave begins a task that is user instance of the class.
f. **Closure of Demons Slaves**: After performing the tasks, can be closed by the demons slaves.
g. **Closure of Master Demon**: Once closed demons slaves, we can close the master daemon.

Master: By creating the master should take into account the implementation of the platform and is declared as follows: private *PlatformBind name*; order to use the functionalities of the platform. The platform has been developed in Java 1.5, from this version has a specialized package for the development of competing sections, the *java.util.concurrent* package. Taken out this is due to declare the interface *ClassStub* is required to be implemented this method in order to pass the request of *PlatformBind,* but in the *execute()* contains the code to execute. It is stored consultation with the role *strQuery(),* and to process information is the role *LeerDatosPreProcessing()*; which obtains documents to carry out the consultation, the format is presented in text mode.

Slave: The slave receives the group and takes the appropriate consultations to carry out the comparison, as the documents are in English-Spanish language must verify the document, this will help to shape what is a single text information in Spanish and English, Formed this is done the PT (Point Transitional) with *getTransitionPoint (String strModified)* where *strModified* is a string that contains documents in English or Spanish. To assess the virtual consultation with paragraph was used according to Jaccard: *Jaccard (String cad1, String cad2)*, which returns a value as a comparison.

7 Results

The next result is the collection of 19 documents in English and Spanish with threshold .005, the results are shown in the following Table 1.

Table 1. Groups formed and assigned to each slave.

Groups	No. Documents	Slaves
0	4	Slave 1
1	4	Slave 2
2	8	Slave 3
3	2	Slave 1
4	1	Slave 2

In the able above were obtained with the 19 documents 5 groups, which were divided between 3 slaves, each processed this information to compare with the query.

The following Table 2 shows the results when comparing the consultation with each group formed and three slaves up.

As can be seen, each slave contains a number of groups, table [I], and each group generates its representation, in both English and Spanish, location of the transition points and then shaping the Virtual Paragraph, the comparison of paragraph Virtual in English and Spanish with the consultation is given in the Table 2.

Subsequently took place with more documents, 40 documents was conducted with both English and Spanish, and were divided into groups each slave to be able to process information and compare it with the consultation.

It is worth mentioning that each slave is generated the point of transition, and it conforms around what is called virtual paragraph will then be compared to the query by applying the role of similarity between Jaccard, the results are shown in the Table 3.

Table 2. It shows the results when comparing the query with each slave, it is worth to say that each slave its rightful certain groups.

Query: Spanish: ataud metal subsuel English: metal conffi subsoil			
PV	**Esclavo 1**	**Esclavo 2**	**Esclavo 3**
Spa	0.0	0.0	0.1176
Eng	0.0	0.0	0.1176
Spa	0.0	0.0	
Eng	0.0	0.0	

Table 3. Groups formed and assigned to each slave.

Groups	No. Documents	Slaves
0	4	Slave 1
1	8	Slave 2
2	12	Slave 3
3	8	Slave 1
4	3	Slave 2
5	4	Slave 3
6	1	Slave 1

As can be seen that the group has more elements should have consulted more widely accepted view, this provides a level of acceptance in the results.

Table 4. It shows the results when comparing the query with each slave, it is worth to say that each slave its rightful certain groups.

Query: Spanish: Campesinos realizan una marcha, ley agraria. English: Peasants perform a march, land law.			
PV	**Slave 1**	**Slave 2**	**Slave 3**
Spa	0.0	0.0	0.03846
Eng	0.0	0.0	0.03333
Spa	0.0	0.0	0.0
Eng	0.0540	0.0	0.0
Spa	0.0	0.0	
Eng	0.0	0.0	

It took time to recover, both linear and in parallel, The results of the linear system shown in Fig. 2 and in parallel in Fig. 3.

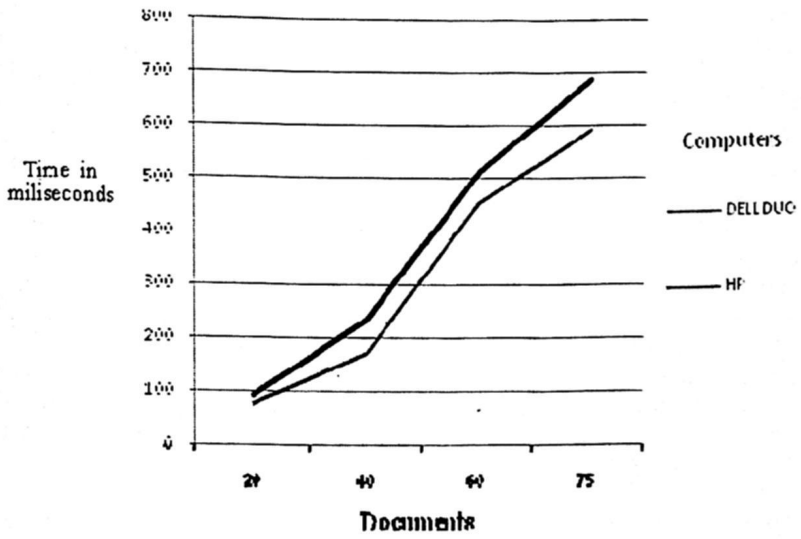

Fig.2. RIP sequential.

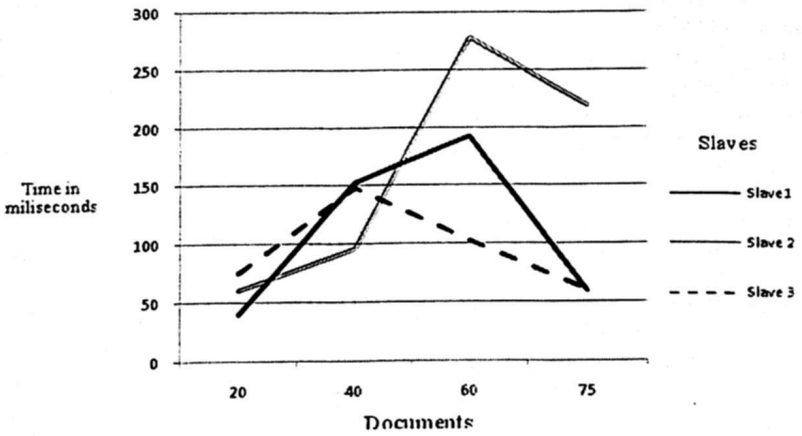

Fig.3. RIP parallel.

As expected, in parallel reduces the time and cost for computer information retrieval.

8 Conclusions

The results so far are satisfactory, work is under way on a platform developed in Java, applications for parallel and distributed [18], the functions of master node are programmed in its entirety. The platform allows for an appropriate way and user-friendly management of shared or distributed memory. The system detects the language of the text and analyze texts in both English and Spanish, also achieves these documents grouped according to criteria such as language and can be seen from the tables above yields a high degree of information retrieval in both languages.

In order to solve problems optimization algorithms using large-scale parallel, it was decided to create a platform itself, aimed directly to exploit the architecture of clusters. One of the most popular tools in science to develop solutions using the mechanism of passing messages is MPI.

The usefulness of Java for concurrence, include a queue asynchronous. With queues asynchronous, is no longer necessary to create critical regions or traffic lights for the addition or subtraction of elements, as they include the control of competition within the same class. The Platform asynchronous queues were used for receiving messages.

Another problem that occurs in the information retrieval recovery is bilingual in both languages, however, for this work was not necessary to translate the whole document but a single paragraph, which significantly reduces the computational cost. In addition the use of a Kraken platform parallel.

References

1. Ángel F. Carlos G. J. Luis. Recuperación de Información utilizando el modelo vectorial. Participación del taller CLEF-2101, Mayo 2002.
2. Van Rijsbergen, C.: "Information Retrieval". Butterworth, London, 1979.
3. Greengrass, E.: "Information Retrieval": A Survey. Tecnical Report, november, 2000.
4. Ruiz Shulcloper, J.; Lazo Cortés, M. ; Alba Cabrera, E.; Pico Peña R.; Sanches Gutierres, I.: "Workshop on Data Mining". Logical Combinatorial Pattern Recognition Group, ICIMAF, Cuba, december, 1999
5. Ester, M.; Kriegel, H.; Sander, J.; Wimmer, M.; Xu, X.:"Incremental Clustering for Mining in a Data Warehousing Enviroment". 1998.
6. Allan, J.; Carbonell, J.; Doddington, G.; Yamrn, J; Yang, Y.: "Topic Detection and Tracking Pilot Study: Final Report". Proceeding of DARPA Broadcast News Transquiption and Understanding Workshop, pp. 204-228, 1998.
7. Nagesh, H.; Goil, S.; Choudhary, A.: "A Scalable Parallel Subspace Clustering Algorithm for Massive Data Set". 2000.
8. Forman, G.; Zhang, B.; "Linear Speedup for a parallel non aprocimate recasting of center based clustering algorithms; including K-Means, K-Harmonics Means and EM", 2000.
9. Gil-García, R. "Paralelización de algoritmos de agrupamientos jerárquicos para semejanzas reducibles en redes de difusión". Tesis de maestría, Departamento de Computación, Universidad Oriente, Cuba, 2000.
10. Sánchez, G.: "Desarrollo de Algoritmos para el agrupamiento de grandes volúmenes de datos mezclados". Tesis de Maestría CIC. IPN. México. 2001.
11. GUERRA A. Aprendizaje Automático: Clasificación, páginas 6-8, 2004.
12. J. L. Neto, A. A. Freitas, and C. A. A. Kaestner. Automatic text summarization using a machine learning approach. In Proceedings of the 16th Brazilian Symposium on Artificial Intelligence, pages 215-225, Porto de Galinhas/Recife, Brazil, 2002.
13. W. T. Chuang and J. Yang. Text summarization by sentence segment extraction using machine learning algorithms. In Proceedings of the 4th Paci c-Asia Conference on Knowledge Discovery and Data Mining. Current Issues and New Applications, pages 454-457, London, UK, 2000.
14. E. Hovy and C.-Y. Lin. Advances in Automatic Text Summarization, chapter Automated text summarization in SUMMARIST, pages 81-94. MIT Press, Cambridge. 1999.
15. S. Teufel and M. Moens. Sentence extraction as a classication task. In Proceedings of the ACL Workshop on Intelligent Text Summarization, pages 58-65, Madrid, España, 1997.
16. K. Rosales-López, M. Tovar-Vidal, D. Vilariño-Ayala, B. Beltrán-Martínez, H. Jiménez-Salazar. "Confección de resúmenes automáticos usando n-gramas" en IEEE 5° Congreso Internacional en Innovación y Desarrollo (Morelos, Cuernavaca). 2007.
17. Manning, D. C. y H. Schütze. Foundations of statistical natural language processing. MIT Press. 1999.
18. Pinacho, E.: "Una plataforma para el desarrollo de aplicaciones en paralelo usando Java" Tesis de Maestría BUAP, México, 2007.