scientific reports

OPEN

A System for Converting and Recovering Texts Managed as Structured Information

Edgardo Samuel Barraza Verdesoto[1,3,4]✉, Marlly Yaneth Rojas Ortiz[3] & Richard de Jesus Gil Herrera[2]

This paper introduces a system that incorporates several strategies based on scientific models of how the brain records and recovers memories. Methodologically, an incremental prototyping approach has been applied to develop a satisfactory architecture that can be adapted to any language. A special case is studied and tested regarding the Spanish language. The applications of this proposal are vast because, in general, information such as text way, reports, emails, and web content, among others, is considered unstructured and, hence, the repositories based on SQL databases usually do not handle this kind of data correctly and efficiently. The conversion of unstructured textual information to structured one can be useful in contexts such as Natural Language Generation, Data Mining, and dynamic generation of theories, among others.

Written communication is a type of information that has a basic structure well defined in each language which is useful in information processing[1]. Additionally, some other types of communication can be totally or partially converted into text whereupon the final processing is carried out[2,3].

Several applications have been developed in information processing following principles or attributes of the text, for instance, to build ontologies or micro-theories[4–7] which are convenient for automatic decision-making tasks[8]. Furthermore, In process automation, the speed at which the information is produced reduces human performance and delays the decision-making processes, this has generated the urgent need to delegate some decisions to machines and create applications for resolving these problems[9–11].

The information retrieval and the generation of natural language promote the generation of sentences/phrases with meaning from a large amount of data generated by interactivity, shared repositories, and homogeneous or heterogeneous data sources, one of these types of application is presented in[12] where is exposed a framework able to generate three classes of question and answers from corpora: *fill in the gaps*, *multiple choice*, and *shuffled sentences*. The framework aims to create a pedagogical tool able to automatically generate tests in the context of a topic, the parser divides texts according to the processed language and prepares the type of question selected. These approaches that allow recording and recovering of fragments or whole texts from a repository, conceiving and improving strategies applied in recovering unstructured information are very important in the current Computer Science.

This article introduces an architecture that allows building applications capable of dissociating texts/sentences in subsets of cores with properties and simple operations such as those that the algebraic groups incorporate. These operations are preferable because they have properties that promote a straightforward and reliable manner to retrieve a whole text or part of it by keeping the structure of the language. Additionally, relationships between subsets are incorporated because they play an essential role in maintaining the meaning of the recovered text/sentences.

The main objective of this manuscript is to show the design of a system with the capability of processing sentences (part of the text), storing them in databases, and finally, recovering them while keeping the original text's basic meaning. To reach that, it is reviewing some previous concepts and experiences about linguistic computational to support the architectural design; it is described and justified how the algebraic groups help in the organization of the components of sentences for storage and recovering them while keeping the meaning and structure; Also, it is treated how to design an architecture for a processing system the objects/data as structured information (into structured databases), and finally, it is shown the functionality of a system for some illustrative instances and test cases for the Spanish language.

[1]Universidad Americana de Europa (UNADE), Cancún, México. [2]Universidad Internacional de la Rioja, Logroño, Spain. [3]Research Department, Tecnológica Autónoma de Bogotá (FABA), Bogotá, Colombia. [4]Universidad de Santander (UDES), Bogotá, Colombia. ✉email: edgardo.barraza@correo.faba.edu.co

This paper is organized as follows. Firstly, a theoretical framework under which the proposal is based will be explored. Secondly, a general architecture will be proposed that incorporates each one of the elements exposed in the theoretical framework. Thirdly, an approach, based on the architecture presented, applied to the Spanish Language will be analyzed. Finally, the findings and future works will be exhibited.

Conceptual framework

Object-action dissociation/integration.

These studies and approaches suggest that the information held in the brain is a set of clusters (cores) that could be affected by the ambiguity and the context in both, the dissociation and integration. Likewise, according to some theories, the brain saved our memories in two ways: semantically and the episodic way[13,14], this latter manner is very important to explain the development of the strategy followed in this paper.

Historically, the dissociation of the information by the human brain was observed when comparing *Broca's aphasic agrammatical patients*, whose speech involves the use of very few verbs in contrast with other *anomic patients* that had great difficulty finding concrete nouns[15]. Initially, the major difficulty with verbs for Broca's patients was interpreted based on the highest syntactical complexity of verbs compared to nouns[16–18]. However, the idea that verbs are, in general, harder to produce has been undermined in other studies where it is indicated that patients with anomic difficulties produce verbs more easily than nouns[19,20]. From a *neurophysiological* point of view, there are different opinions and theoretical proposals[21], of which three hypotheses have been put forward regarding verb-noun storage issues within neural networks: *partial separation of verbs and nouns*[22,23], *word separation based on morphosyntax*[24], and *separation between actions and objects*[25,26]. Psycholinguistics also agree that exist, in the brain, the distinction between various grammatical categories, particularly between verbs and nouns, and propose three possible starting points or context to access the information: *availability of information related to the grammatical class*, *a required grammatical knowledge*, and *the independence between the definition of grammatical class and the semantic differences*[27–30]. A considerable number of studies have dealt with aspects associated with the dissociation of the information within the human mind and the conclusions are similar[31], there are a dissociation between verbs(actions) and nouns (objects).

The counterpart of the dissociation process is the integration process. According to[32], grammatical information is relevant to understanding and producing sentences, but a plausible conclusion suggests that the grammatical class information is not a lexical property that can be retrieved automatically; instead, this property is likely to play an important role in the context of a sentence. Fundamentally, the role of the grammatical class in sentence processing is modulated by the linguistic differences regarding the way as words of certain grammatical classes are used within sentences. In all languages, verbs commonly require higher processing than nouns at various levels, firstly, because the processing of verbs is about events and could exist many elements that will have to be integrated. Secondly, the verb syntax also demands more processing because verbs should be connected to other words to convey their meaning. Lastly, nouns are linked to objects, but they might refer to events too, and it is necessary its disambiguation. In conclusion, the effects of the grammatical class in the retrieval and representation of simple words are more productive when the context is present[33–35].

In addition, Neuroscience states that there are two kind of memory for storing and remembering facts and events consciously; such events are stored in the *episodic memory* such as a storyteller whereas the *semantic memory* records the same event as part of our overall knowledge (dictionary). In specific, the *episodic memory* is intended as a repository in our brain where is recorded an event similar to a text well-written[13,14].

In summary, there are two processes well conceived in our brain, dissociation and integration of an event. These tasks are the fundamentals of the proposal in this paper.

From sentences to clusters of words.

The word classification has been a normal practice in linguistics, computer sciences, and education, among others (see Fig. 1); this practice normally has different targets and results. Furthermore, as an instance, ConceptNet is a project based on the sense common concept that was conceived as a semantic network containing lots of things that the computers should know about the world[36–38]. Another example is the WordNet project which resembles a thesaurus in that words are grouped based on their meanings, the result is a network that can be browseable easily[39–42].

A text is more complex than simple words, it is a texture that relates, firstly, words to create sentences, secondly paragraphs, and ignoring other structures, finally several paragraphs directly or indirectly (e.g. using the anaphor) linked between one another result in a text. Each language has rules to build sentences and paragraphs. According to[43], there are various ways to classify and describe the languages, but a very common is the order of each one of their main components (**Subject**, **Object**, and **Verb**) in the sentence:

- *Subject-Object-Verb* (**SOV**). This is the most frequent type of word order in spoken languages.
- *Subject-Verb-Object* (**SVO**). It is a relevant type of word order because of its speakers worldwide.
- *Verb-Subject-Object* (**VSO**): It represents a relatively small set of languages.
- *Verb-Object-Subject* (**VOS**): Very few languages use this kind of order.

Some approaches use such classifications to divide sentences, expressions, paragraphs, and texts, and, ultimately, to generate categories that are used in specific applications[44,45]. Additionally, other applications use these characteristics in a reverse way, for instance, to build sentences and paragraphs, or concatenate textual expressions from the same or different sources for generating new expressions; this is being applied in Human Machine Interfaces (**HMI**) development[46]. On the other hand, a text not only has nouns and verbs, else other types of words with different purposes, e.g., emphasizing words, which to join small sentences to produce effects like generalization or itemization, etc. These words play an important role to decide how the relations between words, sentences,

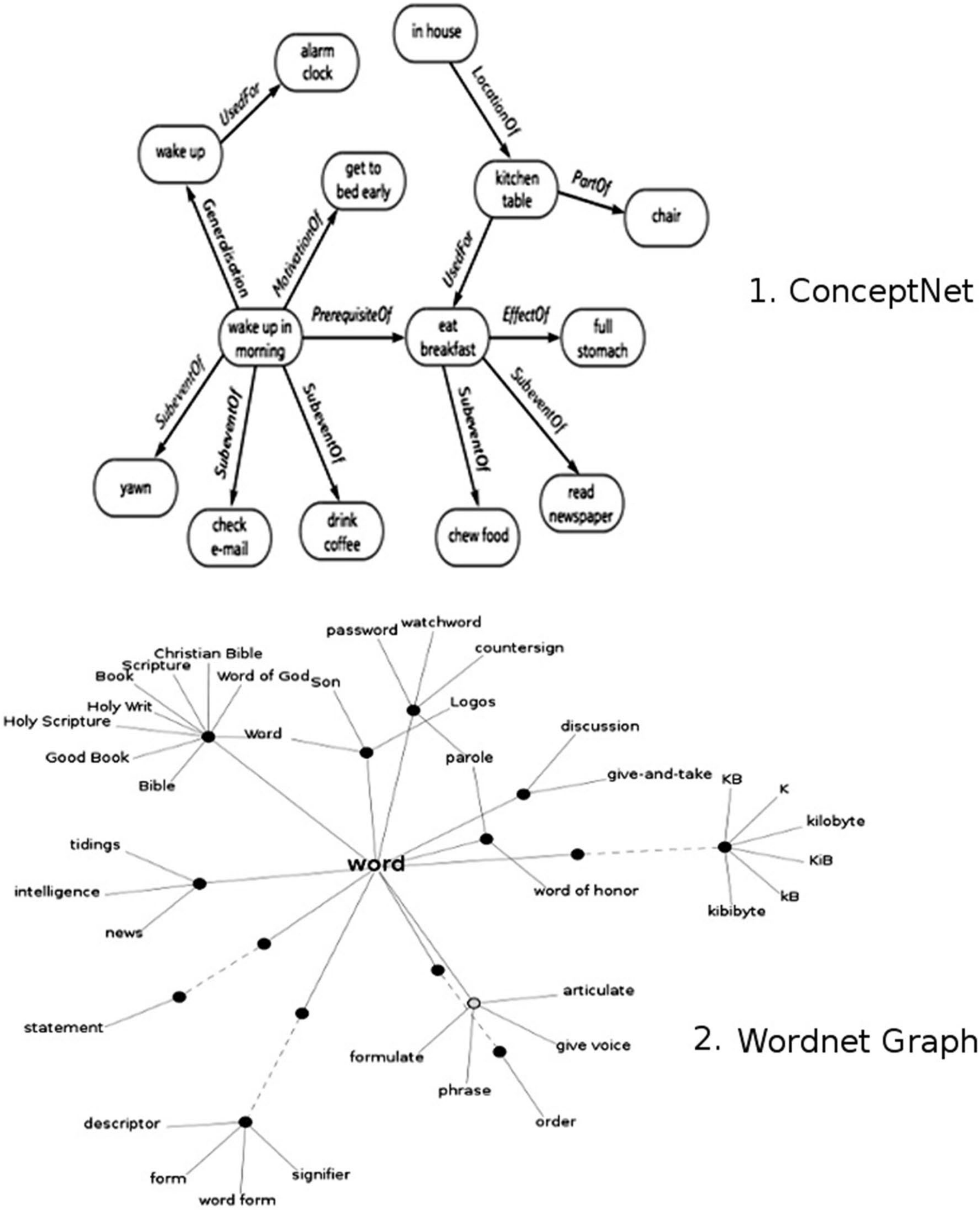

Figure 1. Some word classifications techniques. Source: Extracted and adapted from[37,42].

and paragraphs are. They can be linked to verbs or nouns, e.g. the determinants which comply with the function of generalization or quantification of nouns[47].

In[48] was analyzed the preliminary results focused on the dissociation of sentences in clusters. The sentences studied were in the Spanish Language. Section 2 of this reference exposes why it is necessary to migrate from String-set dependence to another algebraic structure for modeling a sentence, and why this algebraic structure must be an Abelian group, it also supplied the proof. In summary, the dissociation between verbs and nouns, mainly, is a convenient strategy to generate new sentences, also, it is important to create an adequate environment for it.

Algebraic environment. Modern Algebra is a discipline that deals with the properties of the sets and their elements, and the operations that can be executed within them. Modern algebra classifies the sets as **semigroups**,

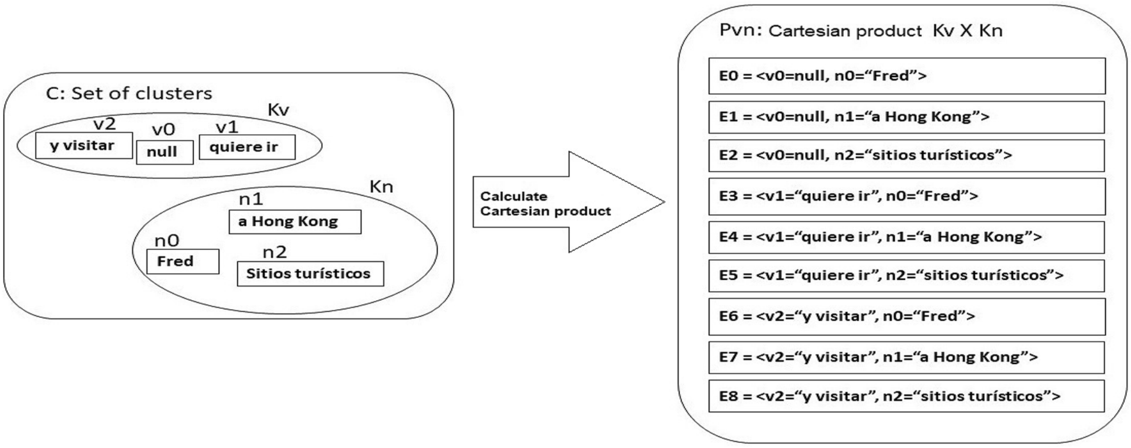
C: Set of clusters
Kv
v2
v0
v1
y visitar
null
quiere ir
Kn
n1
a Hong Kong
n0
Fred
n2
Sitios turísticos
Calculate Cartesian product
Pvn: Cartesian product Kv X Kn
E0 = <v0=null, n0="Fred">
E1 = <v0=null, n1="a Hong Kong">
E2 = <v0=null, n2="sitios turísticos">
E3 = <v1="quiere ir", n0="Fred">
E4 = <v1="quiere ir", n1="a Hong Kong">
E5 = <v1="quiere ir", n2="sitios turísticos">
E6 = <v2="y visitar", n0="Fred">
E7 = <v2="y visitar", n1="a Hong Kong">
E8 = <v2="y visitar", n2="sitios turísticos">

Figure 2. Dissociation of a sentence. Source: Own elaboration.

monoids, **groups**, **rings**, and **fields**; all of these are named algebraic structures. These classifications depend on the number and type of properties that the operation fulfills.

If the elements of sentences are treated like components of an algebraic set, then such components could be used to build phrases and new sentences easily by applying an operation that complies with certain properties. This section shows that converting the conventional algebraic structure of the set of strings (sentences) to a structure more adequate allows for reaching this purpose.

A class very important for this approach is the *groups*, specifically, the **Abelian groups**[49] these last ones have significant properties that guarantee that by operating elements of a dissociated sentence, the original sentence can be rebuilt; A key property is to be commutative because it allows that the result of an operation among elements will be the same, although the operands change their place in the operation.

A sentence could be treated as an ordered set of strings which implies an algebraic structure very simple, but this structure does not is adequate because each string in the sentence complies with a function depending on its position in it, if the sentence is dissociated and then it is reassembled, this last process must guarantee that the product is at least coherent with the structure of the language.

A sentence could be treated as an ordered set of strings which implies an algebraic structure very simple, it is ordered because each string in the sentence complies with a function depending on its position in it, if the sentence is dissociated into strings and later is required its reassembling, this last process must guarantee that the final sentence keeps the structure of the language and its meaning. These conditions comply if the set generated in the dissociation has associated an operation with certain properties which will be shown in this section.

Supposing the following sentence in the Spanish language: *"Fred quiere ir a Hong Kong y visitar sitios turísticos"* (*English meaning: "Fred wants to go to Hong Kong and visit tourist places"*), and it is dissociated in strings with a word each one. One scenario for creating Natural Language from this dissociation will be to use the **conventional algebraic structure of strings** which is composed of the set of strings, and an operator able to join the strings and generate others (closure property). In this algebraic structure the closure property functions as follows:

$$\begin{aligned} \text{String-set} &= \{\text{“Fred″}, \text{“Hong Kong″}, \text{“quiere″}, \text{“ir″}, \text{“a″}, \text{″ y″}, \text{″ visitar″}, \text{″ sitios″}, \text{turisticos}\} \\ \text{A new string} &= \text{“Fred″} + \text{“quiere″} + \text{“ir″} + \text{“a″} + \text{“Hong Kong″} \\ &= \text{“Fred quiere ir a Hong Kong″} \end{aligned}$$

But the closure property is not enough, because the generation of a new string in natural language must ensure structure and meaning, and this is not completely possible in this set with this operation, for example:

$$\begin{aligned} \text{A new string} &= \text{“ir″} + \text{“quiere″} + \text{“a″} + \text{“Fred″} + \text{“Hong Kong″} \\ &= \text{“ir quiere a Fred Hong Kong″(no meaning)} \end{aligned}$$

A possible solution is to divide the sentence into adequate strings forming ordered sets of clusters. The sets generated by this process will be named **Kn** (noun-cores/noun-clusters) and **Kv**(verb-cores/verb-clusters)), but the set used to generate phrases would be the Cartesian product of these sets ($Kv \times Kn = P_{vn}$). In this strategy, the verb-core must contain the null string because the SVO languages (Spanish and English, among others) allow generating phrases without verbs. Figure 2 shows a dissociation following the heuristics in[48]:

The operation should destroy the operand pairs and apply the operation concatenation or plus (+). This method can generate several sentences, but any of them without meaning, or, at least in the context of the original sentence:

Components	Description
$[\![[v_0 : v_1 : \cdots : v_k], [n_0 : n_1 : \cdots : n_k]]\!]$ or $[\![X]\!]$	**Components:** Each component is a vector of vectors. First-internal-vector only contains verbal clusters (v-elements). Second-internal-vector only nominal clusters (n-elements). $[\![X]\!]$ generalizes the internal vectors.
λ_k	$\boldsymbol{\lambda} - element$: Represents a null string. The subscript is the position in the internal vector.
$[\![[\lambda_{0/\infty}], [\lambda_{0/\infty}]]\!]$	$\boldsymbol{\Gamma} - element$ A component with only $\lambda - elements$ in all internal vectors.

Table 1. Set O_{vn} and description of its components. Source: Own elaboration.

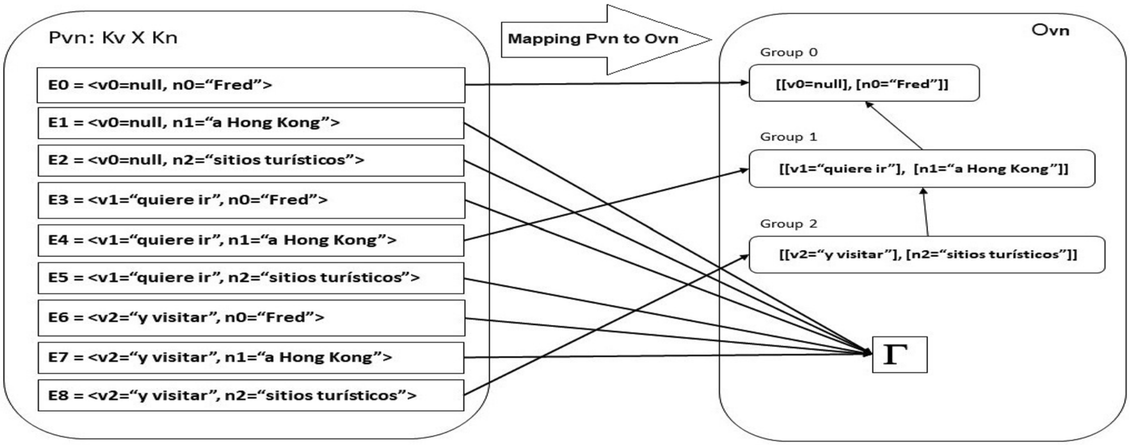
Pvn: Kv X Kn
E0 = <v0=null, n0="Fred">
E1 = <v0=null, n1="a Hong Kong">
E2 = <v0=null, n2="sitios turísticos">
E3 = <v1="quiere ir", n0="Fred">
E4 = <v1="quiere ir", n1="a Hong Kong">
E5 = <v1="quiere ir", n2="sitios turísticos">
E6 = <v2="y visitar", n0="Fred">
E7 = <v2="y visitar", n1="a Hong Kong">
E8 = <v2="y visitar", n2="sitios turísticos">
Mapping Pvn to Ovn
Ovn
Group 0
[[v0=null], [n0="Fred"]]
Group 1
[[v1="quiere ir"], [n1="a Hong Kong"]]
Group 2
[[v2="y visitar"], [n2="sitios turísticos"]]
Γ

Figure 3. Dissociation of a Spanish sentence. Mapping from P_{vn} to O_{vn}. Source: Own elaboration.

$$
\begin{aligned}
\text{E0} + \text{E3} &= \text{null} + \text{“Fred}'' +'' \text{quiere ir a}'' + \text{“Hong Kong}'' \\
&= \text{“Fred quiere ir a Hong Kong}''(\text{correct}) \\
\text{E2} + \text{E4} &= \text{null} + \text{“sitios turisticos}'' +'' \text{quiere ir}'' +'' \text{a Hong Kong}'' \\
&= \text{“sitios turisticos quiere ir a Hong Kong}''(\text{no meaning})
\end{aligned}
$$

So far, this strategy revolves around the *closure property* and other properties such as *associative*, and the *neutral element*; but this is not enough to guarantee structure and meaning, at least compared to the source text. To improve this proposal is necessary to include more properties to the set along with the operation, this is only possible by exploring other possible set types that can build up an algebraic structure more useful, and thus, it is decisive to map P_{vn} to another set that will be named O_{vn}. Table 1 shows the new set and its components.

Mapping $F_{vn} : P_{vn} \rightarrow O_{vn}$: Let us define F_{vn} as:

- Pairs $< v_i, n_j >$ belong P_{vn} with different index $(i \neq j)$ will be mapped to Γ in O_{vn}.
- All pairs mapped must contain at least a *n-element*, then, pairs such as $< v_i, \lambda >$ will be mapped to Γ in O_{vn}.
- Additionally, if couples such as $< v_i, n_i >$ exist in P_{vn}, then, elements such as $< \lambda, n_i >$ will be mapped to Γ in O_{vn}.

Figure 3 show the mapping made from P_{vn} to O_{vn} for the example.

In O_{vn} the operation used, also, change, and it is defined as follows:

- **Dual**. It is *Dual* because of whether two components are operated, then the operation takes place independently in each internal vector. This property allows to separate completely verbs and nouns.
- **Positional**. It is *Positional* because the operation is carried out by two elements with the same subscript. This property allows to implement commutativity.

For example:

$$
\begin{aligned}
&[\![[\lambda_0 : v_1 : \lambda_2][\lambda_0 : n_1 : \lambda_2]]\!] + [\![[\lambda_0 : \lambda_1 : v_2][\lambda_0 : \lambda_1 : n_2]]\!] \\
=&[\![[\lambda_0 + \lambda_0 : v_1 + \lambda_1 : \lambda_2 + v_2][\lambda_0 + \lambda_0 : n_1 + \lambda_1 : \lambda_2 + n_2]]\!] \\
=&[\![[\lambda_0 : v_1 : v_2][\lambda_0 : n_1 : n_2]]\!] \\
=&v_1 + n_1 + v_2 + n_2
\end{aligned}
$$

This operation is the same that:

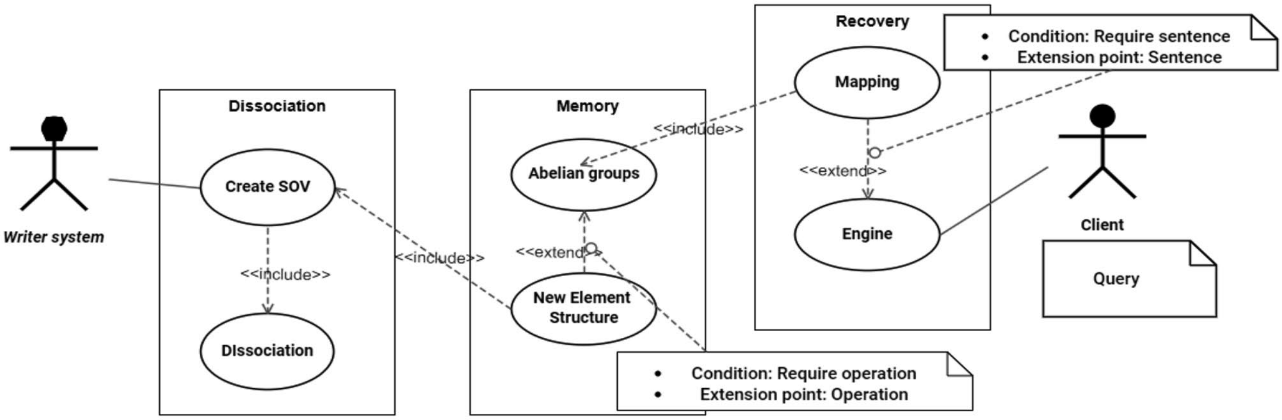

Figure 4. Use case for the architecture of the system. Source: Own elaboration.

$$
\begin{aligned}
&?\llbracket[\lambda_0 :'' \textit{quiere} \quad \textit{ir}'' : \lambda_2][\lambda_0 :'' a \quad \textit{Hong} \quad \textit{Kong}'' : \lambda_2]\rrbracket + \llbracket[\lambda_0 : \lambda_1 :'' y \quad \textit{visitar}''][\lambda_0 : \lambda_1 :'' \textit{sitios} \quad \textit{turisticos}'']\rrbracket?\\
=&?\llbracket[\lambda_0 + \lambda_0 :'' \textit{quiere} \quad \textit{ir}'' + \lambda_1 : \lambda_2 +'' y \quad \textit{visitar}''][\lambda_0 + \lambda_0 :'' a \quad \textit{Hong} \quad \textit{Kong}'' + \lambda_1 : \lambda_2 +'' \textit{sitios} \quad \textit{turisticos}'']\rrbracket?\\
=&?\llbracket[\lambda_0 :'' \textit{quiere} \quad \textit{ir}'' :'' y \quad \textit{visitar}''][\lambda_0 :'' a \quad \textit{Hong} \quad \textit{Kong}'' :'' \textit{sitios} \quad \textit{turisticos}'']\rrbracket?
\end{aligned}
$$

To create the sentence starting from this new core is applied a process that states that for each position in the vectors a verb is concatenated with the noun corresponding and the result will be added to the next result as follows:

$$
\begin{aligned}
&='' (\textit{quiere} \quad \textit{ir}'' +'' a \quad \textit{Hong} \quad \textit{Kong})''\\
&='' \textit{quiere} \quad \textit{ir}'' +'' a \quad \textit{Hong} \quad \textit{Kong}'' +'' (y \quad \textit{visitar}'' +'' \textit{sitios} \quad \textit{turisticos})''\\
&='' \textit{quiere} \quad \textit{ir} \quad a \quad \textit{Hong} \quad \textit{Kong} \quad y \quad \textit{visitar} \quad \textit{sitios} \quad \textit{tursticos}''
\end{aligned}
$$

It is too easy to deduct that this new operation in O_{vn} is commutative, i.e., the result is the same, although the operands will change their position. This commutative structure is known as an **Abelian monoid structure** and, in[48], and by including the **symmetrical element**, is converted to an Abelian group.

An algebraic structure as has been defined is very useful because reduces the complexity in the reconstruction of phrases because the operation is easy to implement and its behavior is similar to the add operation in the numbers by managing sentences as sets of discrete cores. In the section ”An approach” will be explained that a sentence can generate several Abelian groups, and each one can generate sentences separately.

Methodology

This section proposes an architecture of a system, for dissociating and recovering texts and sentences, based on the concepts, theories, and regulations aforementioned. Figure 4 shows a scheme of the system based on use cases view[50]. The system would include three major sub-systems: **dissociation**, **memory** and **recovery**. The two first sub-systems will be activated serially and immediately after a reading takes place, and the latter process is executed when a query promotes the generation of sentences. Nevertheless, in terms of the information processing associated with each sub-system, they operate independently. The entire system is conceptualized as a framework that could be up-gradable and enriched with plug-in modules.

The class diagram is shown in Fig. 5.

And the activities diagram is shown in Fig. 6, this last diagram is only for dissociating, because the recovery depends on the implementation which is shown in Section ”An approach”.

Dissociation subsystem.

The function of the dissociation subsystem is to split a text/sentence into special units. As previously mentioned in section *From sentences to clusters of words*, all languages share a common characteristic which is the identification of three basic clusters within a sentence: *Subject*(**S**), *Verb*(**V**), and *Object*(**O**). They can occur within a sentence in a different order depending on the language.

In this paper, the expression *SOV-trio* or simply *SOV* will be used to represent the trio that models a sentence or a text. Given that *Subject* and *Object* have similarities both will be treated as (**S**). Additionally, each of the components of a *SOV* will be named a *core*.

The cores may contain one or more words from the sentence. For example, it is possible to have a verb followed by another verb in the same core, as in the following sentence: *”Fred quiere ir a Hong Kong y visitar sitios turísticos”* the two verbs (”quiere ir”) constitute a *V* core. Once a *SOV* is generated, this is dispatched to the memory subsystem.

Strategies to generate SOVs

As explained in section *Object-action dissociation/integration*, there is a consensus about the dissociation between actions (verbs) and objects (nouns) inside the human mind. However,[32] emphasizes the existence of problems by establishing the grammar category that can generate confusion between verbs and nouns, this also

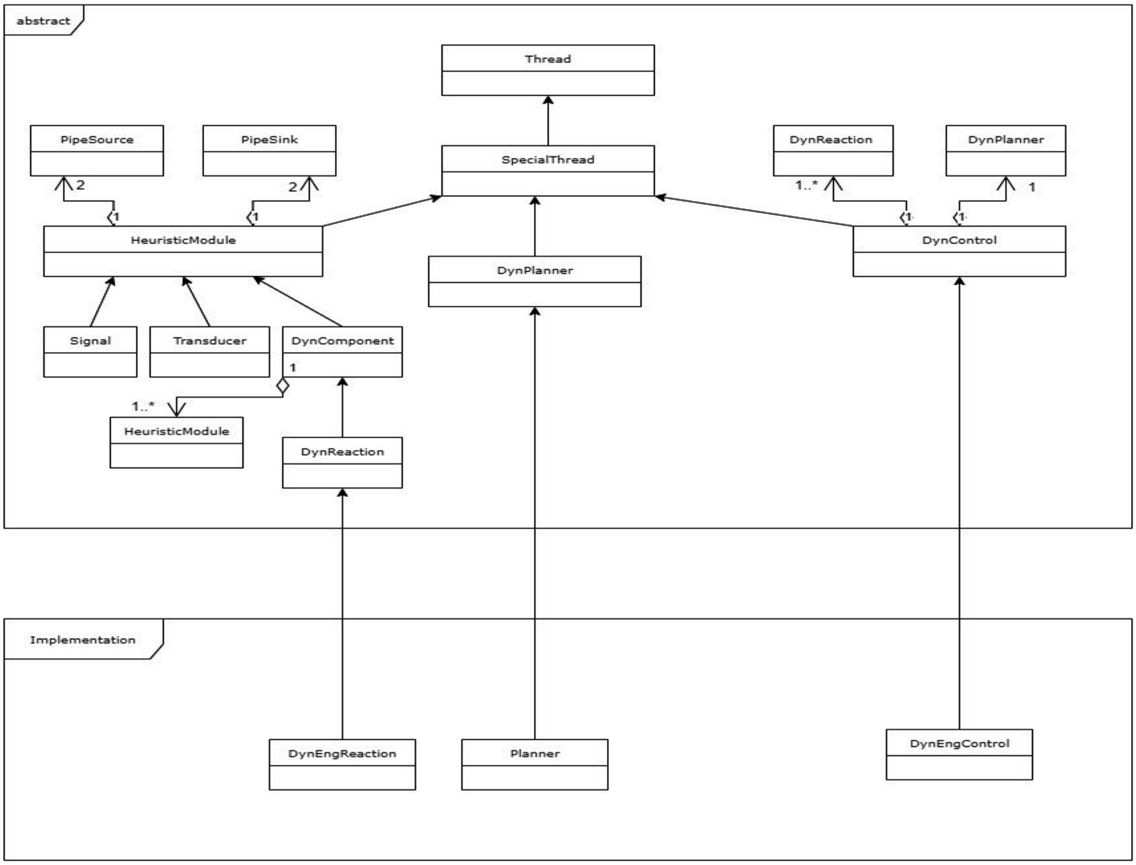

Figure 5. Class diagram of the framework. Source: Own elaboration.

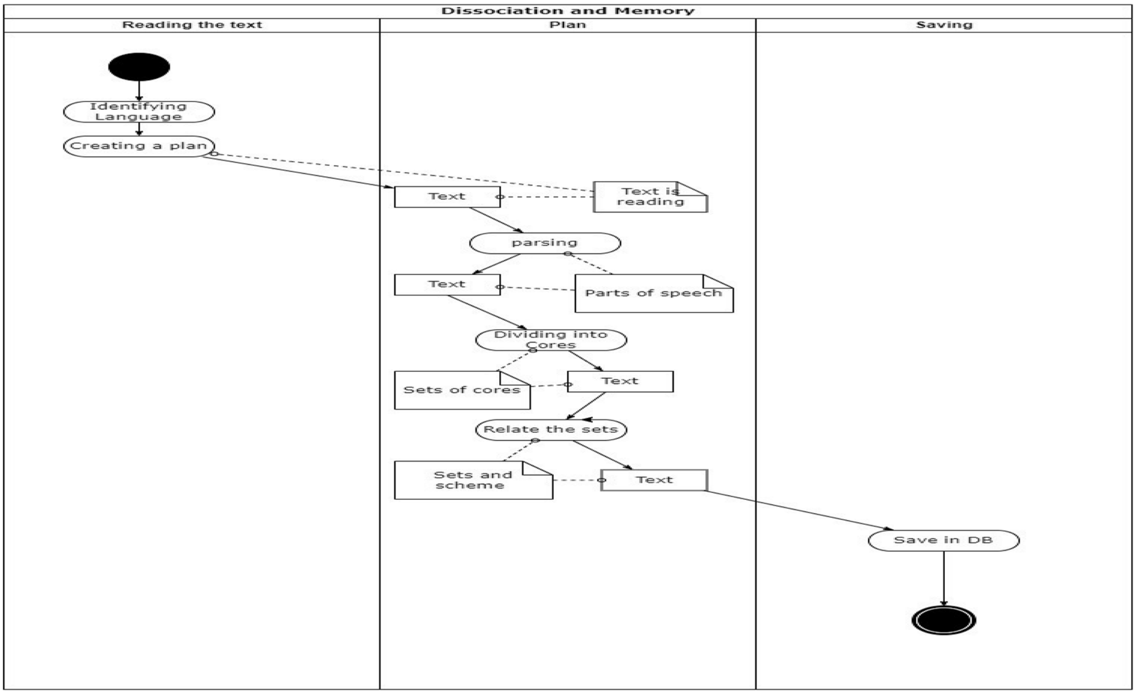

Figure 6. Activities diagram of the framework. Source: Own elaboration.

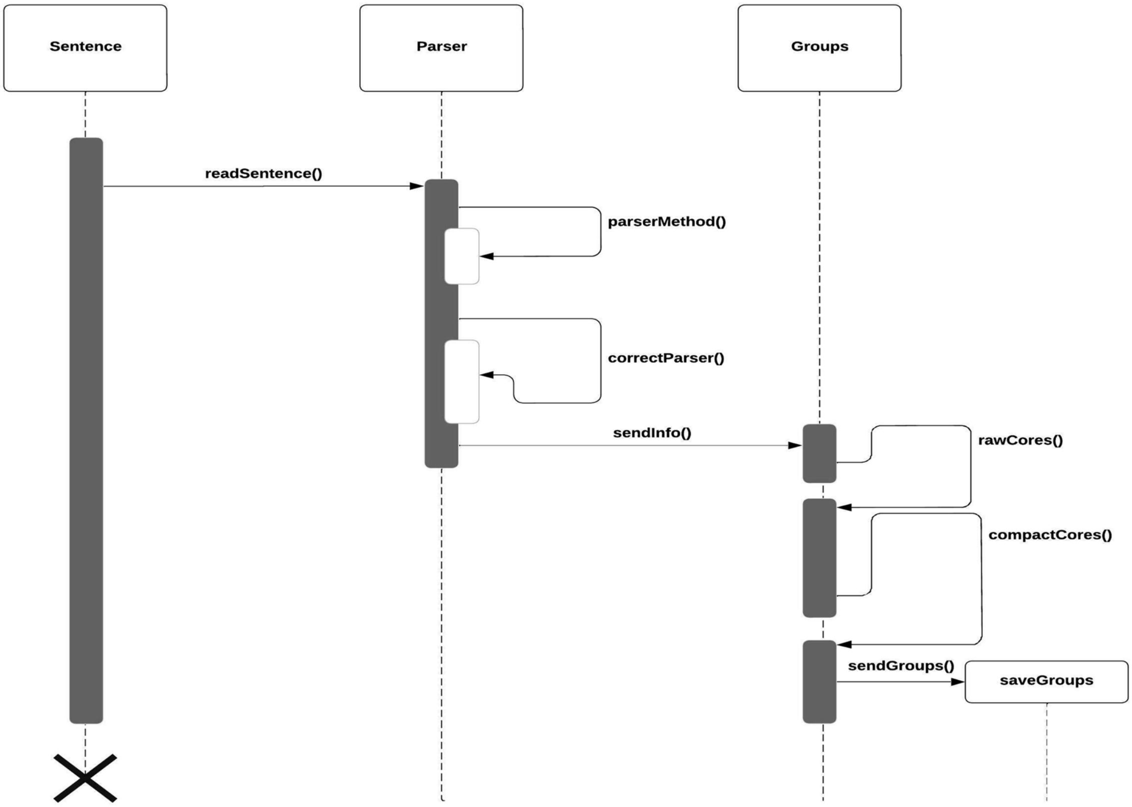

Figure 7. A generic UML sequence diagram for the *dissociation process*. Source: Own elaboration.

can happen in the process of dissociation in this subsystem. To dissociate the sentences correctly, the subsystem should implement modules such as:

- *Syntactic Analysis (Parsing)*. An ordinary parser generates a syntax tree from which the *SOVs* can be rapidly built. Although this strategy is good, it does not avoid that the syntax tree generated may require the involvement of some other heuristic processes to "refine" the creation of the cores, for instance, in cases of slang interpretation as is shown in Fig. 7, **Parser module** of, routine **parserMethod()**.
- *Dictionaries and conjugators*. Sometimes, parsers can produce an incorrect word classification, especially when the parser has not well-trained in a particular language, in such case it is necessary to perform an analysis and debugging process over these words. For this purpose, software like dictionaries and conjugators modules could be useful to validate the category as is shown in Fig. 7, **Parser module**, routine **correctParser()**.
- *Grouping of elements*. The dissociation in cores requires identifying elements like *determinants*, *adverbs*, *prepositions*, *conjunctions*, etc., in such a way that they will be inserted in the adequate core. This process should be customized for each language as is shown in Fig. 7, **Groups module**.

To summarize, some procedures, syntactic-semantic strategies, and heuristics should be implemented to help in building the *S/O/V cores* correctly.

Memory subsystem.

An important function of the memory system is to store the information generated by the dissociation subsystem. Hence, it is mandatory to build a structure that guarantees order and efficiency. Therefore, the memory system should contain a repository to save the *SOVs* generated by each text read interrelated between them. This storage should maintain these cores in such a way that can be retrieved in the exact order as they were read. According to these principles, the implementation should comply with the following conditions:

- **SQL-database**. The type of database towards has been addressed in this research is the SQL-database because it is the most used to store information. The main idea is to save the texts in such a way that their elements will be organized in groups or clusters representing sets that, joining them, can reproduce the source without losing their meaning.
- **Repository based on queries of cores**. Firstly, a repository based on query means that uses SQL technology to save and recover information. Secondly, the queries can be attended by modules that recover cores, compare against the queries, determine similarity, and create sentences, paragraphs, and full texts as of the cores chosen.

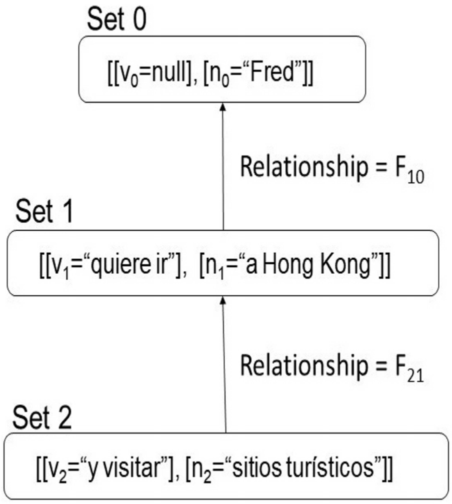
Set 0
[[v0=null], [n0="Fred"]]
Relationship = F10
Set 1
[[v1="quiere ir"], [n1="a Hong Kong"]]
Relationship = F21
Set 2
[[v2="y visitar"], [n2="sitios turísticos"]]

Figure 8. Relationship between groups generated by the sentence: "Fred quiere ir a Hong Kong y visitar sitios turísticos". Source: Own elaboration.

- **Meta-engine**. Each implementation should program a meta-engine that works over the database in a superior layer that the database engine, this should be equipped with the algebraic operation explained in subsection *Algebraic environment* and mappings between groups to integrate them and build a part or whole original text.

Dynamic structure

The dissociation in *SOVs* and the mapping create sets distributed and connected in terms of their original semantic content. Figure 8 shows a scheme that illustrates the relationship between the sets of *SOVs* (*Abelian group*).

The nodes will be related with adequate functions (mapping) to guarantee that the recovery of the part, or the whole, of a sentence/text will be executed correctly as will be explained later.

Recovery subsystem.

The purpose of this subsystem is to generate, in a dynamic way, a sentence/text part or entirely. This subsystem is closely interrelated to the *dynamic structure* because this subsystem is composed of the functions that connect the nodes.

The Engine

The queries are expected in natural language and it would transform into a set of *SOVs*. The key is to compare *SOVs* for finding the closest results. The strategies to match the *SOVs*.can be wide. An example could be to establish matches of *SOVs* that contain elements that could respond contextually to the query as in Fig. 9. The degree of coincidence will be the measure.

This strategy could recover sentences that do not answer the query completely, hence, it would be important to implement another stage. For instance, that compares sentences in a logical context. This can be carried out by converting the query and the text recovered into small text-theories that can be matched logically.

An approach

This section has a summary of a prototype designed as a **layered framework** that could be used for any language characterized as S-V-O (Spanish, English, etc.) The most relevant layers of the dissociation processes are the following:

1. **Identifying the language**. This first layer has been designed to identify the language of the text and divide it into sentences, and finally, their results will send to the next layer one at a time.
2. **Planning**. This second layer chooses the modules required to dissociate the sentences based on the language recognized. This layer makes flexible the framework because it allows changing the rules of dissociation depending on the language to be processed.
3. **Reaction layer**. This layer is related to the strategies to generate *SOVs* which were described in the paragraph *Strategies to generate SOVs*. Figure 7 is shown the execution of two modules in a pipeline way, but new modules could be included to improve the results, this will depend on the implementation. The name of this layer is due to the modules chosen by the *plan layer* being triggered dynamically and executed like a chain reaction in a pipeline.

In this implementation, the modules created in the reaction layer dealt with sentences in the Spanish language (S-V-O language) and were organized in three linear phases following the guidelines described in the paragraph

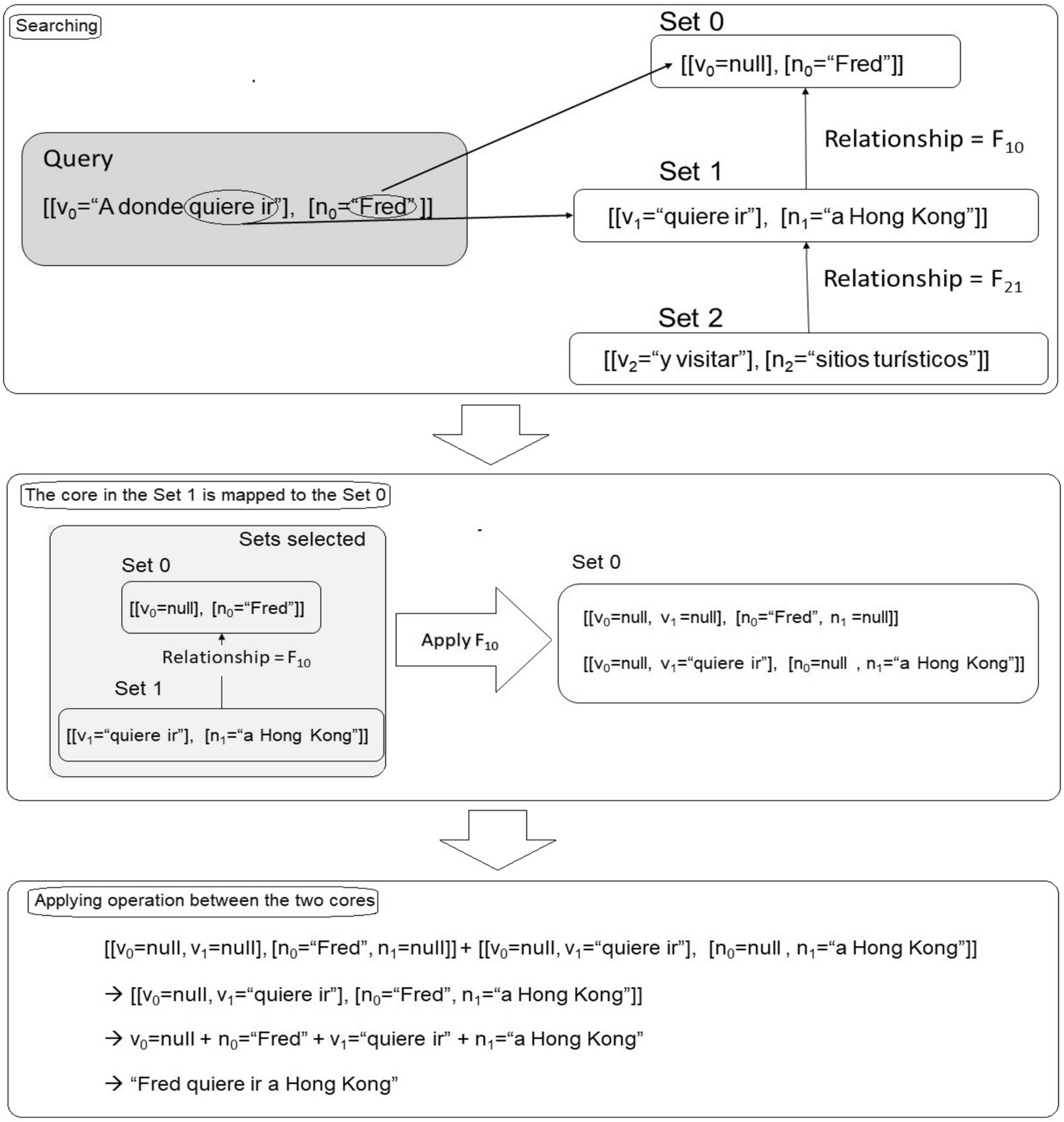
Searching
Set 0
[[v_0=null], [n_0="Fred"]]
Relationship = F_{10}
Query
[[v_0="A donde quiere ir"], [n_0="Fred"]]
Set 1
[[v_1="quiere ir"], [n_1="a Hong Kong"]]
Relationship = F_{21}
Set 2
[[v_2="y visitar"], [n_2="sitios turísticos"]]
The core in the Set 1 is mapped to the Set 0
Sets selected
Set 0
[[v_0=null], [n_0="Fred"]]
Relationship = F_{10}
Set 1
[[v_1="quiere ir"], [n_1="a Hong Kong"]]
Apply F_{10}
Set 0
[[v_0=null, v_1 =null], [n_0="Fred", n_1 =null]]
[[v_0=null, v_1="quiere ir"], [n_0=null , n_1="a Hong Kong"]]
Applying operation between the two cores
[[v_0=null, v_1=null], [n_0="Fred", n_1=null]] + [[v_0=null, v_1="quiere ir"], [n_0=null , n_1="a Hong Kong"]]
→ [[v_0=null, v_1="quiere ir"], [n_0="Fred", n_1="a Hong Kong"]]
→ v_0=null + n_0="Fred" + v_1="quiere ir" + n_1="a Hong Kong"
→ "Fred quiere ir a Hong Kong"

Figure 9. Creating a sentence from a query. Source: Own elaboration.

Strategies to generate SOVs and shown in Fig. 7. In the first phase, each sentence is processed by a linguistic tool, commanded by the VISL parser[51] *reaction layer*, in this stage, it, also, corrects possible inconsistencies generated by the parser as the wrong classification of the words, e.g., some words classified as nouns or vice versa. The information produced by the parser is significant, therefore, it is discriminated, and sent to the next module in the pipeline. The second phase receives the information and classification and applies heuristics for generating, initially, *raw-clusters*, then refined by another heuristic, and finally to produce the set O_v. Lastly, in the third phase, the set O_v is saved in a standard database (SQL-style). The heuristics applied in this approach are not extendable to other languages. However, currently, they are being tested in the English Language, also S-V-O language, to measure their effectiveness in it. Each sentence is organized in Abelian groups hierarchically organized with a binary operation capable of building phrases (see Fig. 3). The Abelian groups obey the specifications done in section *Algebraic environment*.

Table 2 shows the classification established heuristically for the Spanish language cores in this approach. This process involves a loop where neighboring words that comply with certain conditions are packed into a single class named: **nominal core (S)**, **determinant**, and **verbal core (V)**. A *determinant* is used to interrelate Abelian groups as in Fig. 8 in the paragraph *Dynamic structure*. It is important note that the punctuation signs are useful to create these categories, some are part of the **determinants** and other are par of the **verbal cores** or **nominal cores**, for example, in Fig. 8 the **nominal core** n_6 in G_2 include a comma: ", la Paz y la justicia", similarly, the **verbal cores** v_5 in G_3: ", recibía".

Category	Description
p, J	*Determinants*
v, V	*Verbal core*
n	*Noun core*

Table 2. Final categorization scheme.

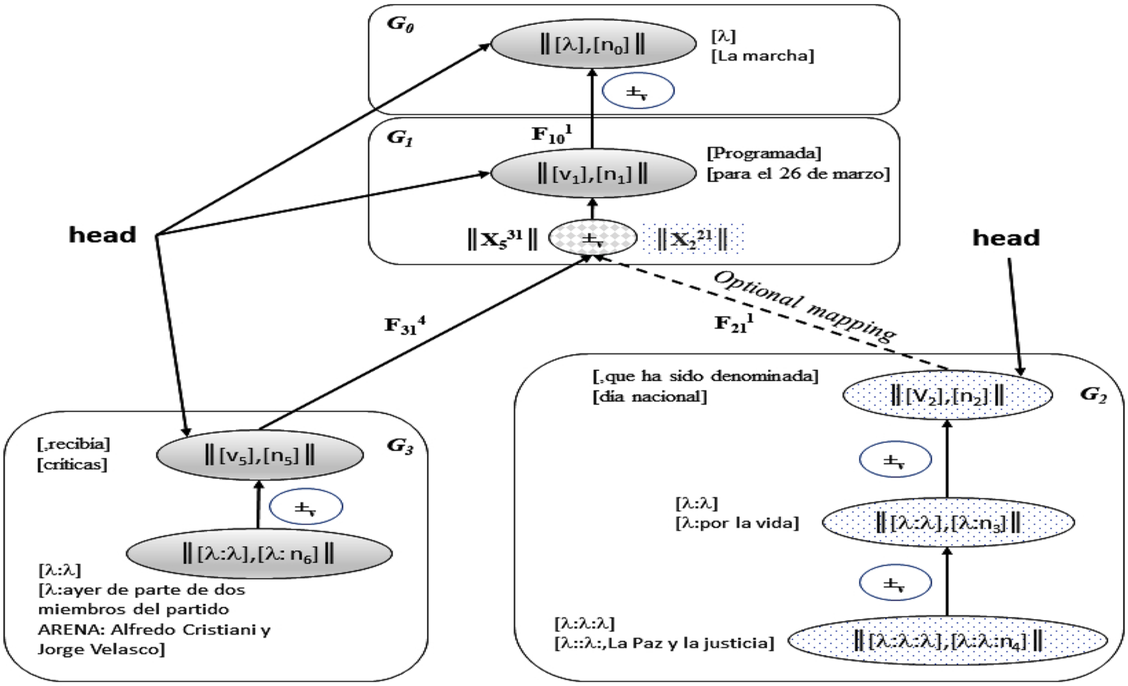

Figure 10. Scheme to recover sentences. Source: Extracted from[52].

In this approach, there are two types of determinants *p-det* and a *J-det*; both interrelate the sets with functions, but the Abelian group pointed by a *J-det* is considered optional in the rebuild of the sentence. All of these properties were established empirically.

The restoring process is not the reverse operation exactly, else it is a complex process that executes tasks from the repository trying to preserve syntax and the original semantics. This purpose is successful due to the properties of the Abelian groups (see[52]) and the hierarchy of sets created by the determinants in the dissociation process. The process is shown in Fig. 10

The sets are operated and mapped in a domino way from the core where the matching occurs up until the root of the hierarchy. For example, in Fig. 10, if the core matched is in G_3 and corresponds to ”n_2 = críticas” then the recovered sentence will be: “*La marcha programada, para el próximo 26 de marzo, recibía críticas*”.

Proposal comparison

Currently, generating sentences and small texts is a task very significant in several fields of Computer Science. The approach named **Rhetorical Structure Theory (RST)**[53] is one of the first proposals created to divide the discourse and has been the inspiration for **Natural Language Generation** (**NLG**) schemes.

The strategy used in *RST* to divide the texts into cores and organize them hierarchically (nucleus and satellite) requires training of neural networks[54,55]. The relevance of the framework presented in this document is that it does not need training.

With respect to the generation of sentences, the implementations and approaches are very exigent. According to[56], they should carry out several complicated tasks as below:

1. Determining the information relevant. This part is associated with the context and scope, basically related to the searching[57–60].
2. Determining the order as the words should appear. Some approaches try to resolve this part from texts by collecting, recovering, and organizing sentences inside them[61,62]
3. Determining: how should be the information aggregated? This stage is considered very difficult because the information can be provided by several sources or it is not the correct response to any query. Some works use the context to resolve discrepancies or the domain to explore the sources[63,64].

4. Determining the right words and phrases (verbs and nouns). This part contains two stages but they will be joined in one because can be carried out jointly. In this part the sentence is organized in one of the following structures: SVO, SOV, VSS, VOS, also, it is analyzed the verb times[65].
5. Combining words and phrases to generate well-formed sentences. This phase builds the sentences, sometimes through templates, or grammar-based techniques, among others[66,67].

In the framework exposed in this document, the first three steps are part of the recovery system in this framework, specifically, corresponding to the engine searching. the last two steps can be resolved by responding to the queries and executing the algebra of the groups and mapping between them which are tasks easy and efficient. All of these show a framework simple to implement.

Conclusions and future works

The high demand for information has caused an increasingly important in the automation of processes such as decision-making, pattern recognition, and interaction human-machine, among others. Several of these processes require the use of the text, either to understand queries, generate reports, or answer in natural language, hence, building applications with these functions takes a greater relevance. This paper presents an architecture for dissociating the text/sentences, saving it in a SQL database, and recovering it without loss of meaning. This is highly productive in process automation because the textual information is converted from unstructured to structured format and the queries and other processes in natural language can be more efficient.

The suggested system has been inspired and based on processes verified by scientists related to the dissociation of the information inside the human brain, memory models in the Neuroscience field, and the structure of the languages in Linguistic and Psycholinguistic disciplines. The proposed framework divides a sentence/text into clusters like the brain dissociates the speech into nominal and verbal categories. The scheme will divide the text/sentence into sets of cores named *nominal cores* and *verbal cores*, and implement an algebraic operation that can be used to generate new sentences that keep the original meaning without loosing the structure of the language. This proposal was applied by the approach studied in the last section successfully.

The explored implementation resolved a great part of the challenges described in the paper by implementing a framework with abstract modules that can be custom implemented, for instance, the processes described in the architecture, the generic abstract modules for different languages, and the recovery modules, among others.

Additionally, the implementation creates a solution for the Spanish language by using heuristics for both dissociation and recovery processes. The application suggests interrelating the algebraic sets by employing functions to recover the whole or part of the textual information by maintaining the meaning. The approach shows that for the Spanish language is possible to have an implementation. In[68] is exposed several proposals of NLG.

A system has been proposed for converting unstructured textual information to be computationally managed structured information. This proposal has been tested in an approach for the Spanish language successfully. Future works will be addressed to implement this framework for other languages and to generate applications for these approaches.

Data availability

The current document has been focused on the discussion about a framework able to compose strategies to divide sentences and texts into cores to save them in a SQL-databases. Under this context the data used to verify the effectiveness belong to other work where the purpose was to study heuristics for dividing the sentences into these clusters, these data do not form part of the current research. Therefore, all data generated or analyzed during this study are included in this published article.

Received: 1 February 2022; Accepted: 13 December 2022
Published online: 23 December 2022

References

1. D'Souza, S. Parser extraction of triples in unstructured text. arXiv preprint arXiv:1811.05768 (2018).
2. YL. Shuea, C. V., P. Keatingb & Yub, K. Voicesauce: A program for voice analysis. In *INTERNATIONAL CONGRESS OF PHONETIC SCIENCES (ICPhS, XVII)*, 1846–1849 (2011).
3. Jain, A. K. & Yu, B. Automatic text location in images and video frames. *Pattern Recogn.* **31**, 2055–2076 (1998).
4. Cimiano, P. & Völker, J. Text2onto - a framework for ontology learning and data-driven change discovery. In *Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB)*, 227–238 (Alicante, Spain, 2005).
5. Ghidini, C. & Giunchiglia, F. Local models semantics, or contextual reasoning=locality+compatibility. *Artif. Intell.* **127**, 221–259. https://doi.org/10.1016/S0004-3702(01)00064-9 (2001).
6. Guha, R. Contexts: A formalization and some applications (1992).
7. Lenat, D. B. & Guha, R. V. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project* (Addison-Wesley Longman Publishing Co., Inc, USA, 1989).
8. Herrera, R. J. G. & Martin-Bautista, M. J. A novel process-based kms success framework empowered by ontology learning technology. *Eng. Appl. Artif. Intell.* **45**, 295–312 (2015).
9. Abbes, H. & Gargouri, F. Mongodb-based modular ontology building for big data integration. *J. Data Semant.* **7**, 1–27 (2017).
10. Gruber, T. R. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.* **43**, 907–928. https://doi.org/10.1006/ijhc.1995.1081 (1995).
11. Guo, K. & Ma, J. Semantic-based heterogeneous multimedia big data retrieval. In Li, K., Jiang, H., Yang, L. T. & Cuzzocrea, A. (eds.) *Big Data - Algorithms, Analytics, and Applications*, 18 (Chapman and Hall/CRC, New York, 2015).
12. Perez, N. & Cuadros, M. Multilingual call framework for automatic language exercise generation from free text. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 49–52 (2017).

13. Tulving, E. Episodic and semantic memory. In Tulving, E. & Donaldson, W. (eds.) *Organization of Memory*, 381–403 (Academic Press, New York, 1972).
14. Havel, I. M. *Strategies of Remembrance: From Pindar to Hölderlin*, chap. 2 (Cambridge Scholars, England, 2009).
15. D. C. Bradley, M. F. G. & Zurif, E. B. Syntactic deficit in broca's apahsia. In Caplan, D. (ed.) *Biological studies of mental processes*, vol. 14, 345–360 (MIT Press, 1988).
16. Saffran, E. M. Neuropsychological approaches to the study of language. *Br. J. Psychol.* **73**, 317–337. https://doi.org/10.1111/j.2044-8295.1982.tb01815.x (1982).
17. E. M. Saffran, M. F. S. & Marin, O. S. M. The word order problem in agrammatism: Ii. production. *Brain and Language***10**, 263—280, https://doi.org/10.1016/0093-934X(80)90056-5 (1980).
18. Feng, S., Qi, R., Yang, J., Yu, A. & Yang, Y. Neural correlates for nouns and verbs in phrases during syntactic and semantic processing: An fmri study. *Journal of Neurolinguistics* **53**, 100860 (2020).
19. Miceli, G., Silveri, M. C., Nocentini, U. & Caramazza, A. Patterns of dissociation in comprehension and production of nouns and verbs. *Aphasiology* **2**, 351–358. https://doi.org/10.1080/02687038808248937 (1988).
20. Miceli, G., Silveri, M. C., Nocentini, U. & Caramazza, A. On the basis of the agrammatics' difficulty in producing main verbs. *Cortex* **20**, 207–220. https://doi.org/10.1016/S0010-9452(84)80038-6 (1984).
21. Zheng, W. *et al.* Chasing language through the brain: Successive parallel networks. *Clin. Neurophysiol.* **132**, 80–93 (2021).
22. Damasio, A. R. & Tranel, D. Nouns and verbs are retrieved with differently distributed neural systems. In *Proceedings of the National Academy of Sciences U.S.A.*, vol. 90, 4957-4960, https://doi.org/10.1073/pnas.90.11.4957 (1993).
23. Daniele, A., Giustolisi, L., Silveri, M. C., Colosimo, C. & Gainotti, G. Evidence for a possible neuroanatomical basis for lexical processing of nouns and verbs. *Neuropsychologia* **32**, 1325–1341. https://doi.org/10.1016/0028-3932(94)00066-2 (1994).
24. K. A. Shapiro, L. R. M. & Caramazza, A. Cortical signatures of noun and verb production. In *Proceedings of the National Academy of Sciences U.S.A.*, vol. 103, 1644—1649, https://doi.org/10.1073/pnas.0504142103 (2006).
25. Siri, S. *et al.* The neural substrate of naming events: Effects of processing demands but not of grammatical class. *Cereb. Cortex* **18**, 171–177. https://doi.org/10.1093/cercor/bhm043 (2008).
26. Tyler, L. K. & Marslen-Wilson, W. Fronto-temporal brain systems supporting spoken language comprehension. *Philos. Trans. R. Soc. B* **363**, 1037–1054. https://doi.org/10.1098/rstb.2007.2158 (2008).
27. Levelt, W. J. M. Speaking: From intention to articulation. *The American Journal of Psychology* (1990).
28. Garrett, M. F. Syntactic processes in sentence production. In R. J. Wales, E. W. (ed.) *New Approaches to Language Mechanisms*, vol. 12, 231—255 (North Holland Publishing Company, Netherlands, 1976).
29. Garrett, M. F. The organization of processing structure for language production: applications to aphasic speech. In D. Caplan, A. S., A. R. Lecours (ed.) *Biological Perspectives on Language*, vol. 12, 172—193 (The MIT Press, 1984).
30. Ullman, M. T. *et al.* A neural dissociation within language: Evidence that the mental dictionary is part of declarative memory and that grammatical rules are processed by the procedural system. *J. Cognitive Neurosci.* **9**, 266–276. https://doi.org/10.1162/jocn.1997.9.2.266 (1997).
31. Elli, G. V., Lane, C. & Bedny, M. A double dissociation in sensitivity to verb and noun semantics across cortical networks. *Cereb. Cortex* **29**, 4803–4817 (2019).
32. Vigliocco, G., Vinson, D. P., Druks, J., Barber, H. & Cappa, S. F. Nouns and verbs in the brain: A review of behavioural, electrophysiological, neuropsychological and imaging studies. *Neurosci. Biobehav. Rev.* **35**, 407–426. https://doi.org/10.1016/j.neubiorev.2010.04.007 (2011).
33. Blything, L. P. & Cain, K. The role of memory and language ability in children's production of two-clause sentences containing before and after. *J. Exp. Child Psychol.* **182**, 61–85 (2019).
34. Khader, P., J. S., Scherag, A. & Rösler, F. Differences between noun and verb processing in a minimal phrase context: A semantic priming study using eventrelated brain potentials. *Cognitive Brain Research***17**, 293—313, https://doi.org/10.1016/S0926-6410(03)00130-7 (2003).
35. Gomes, W., Ritter, V. C., Tartter, H. G., Vaughan, J. R. & Rosen, J. J. Lexical processing of visually and auditorily presented nouns and verbs: evidence from reaction time and n400 priming data. *J. Cogn. Neurosci.* **6**, 121–134. https://doi.org/10.1016/S0926-6410(97)00023-2 (1997).
36. Rodosthenous, C. *et al.* Using crowdsourced exercises for vocabulary training to expand conceptnet. In *Proceedings of The 12th Language Resources and Evaluation Conference*, 307–316 (2020).
37. Singh, H. L. P. Conceptnet - a practical commonsense reasoning tool-kit. *BT Technol. J.* **22**, 211–226. https://doi.org/10.1023/B:BTTJ.0000047600.45421.6d (2004).
38. Speer, R. & Havasi, C. Representing general relational knowledge in conceptnet 5. In *LREC*, 3679–3686 (European Language Resources Association (ELRA), Istanbul, Turkey, 2012).
39. Vial, L., Lecouteux, B. & Schwab, D. Sense vocabulary compression through the semantic knowledge of wordnet for neural word sense disambiguation. arXiv preprint arXiv:1905.05677 (2019).
40. Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D. & Miller, K. J. Introduction to wordnet: An on-line lexical database. *Int. J. Lexicogr.* **3**, 235–244. https://doi.org/10.1093/ijl/3.4.235 (1990).
41. Fellbaum, C. English verbs as a semantic net. *Int. J. Lexicogr.* **3**, 278–301. https://doi.org/10.1093/ijl/3.4.278 (1990).
42. Group, P. F. Visualization of wordnet using force-directed graphs (1991).
43. Boeree, G. Basic language structures.
44. García-Méndez, S., Fernández-Gavilanes, M., Costa-Montenegro, E., Juncal-Martínez, J. & González-Castaño, F. J. A library for automatic natural language generation of spanish texts. *Expert Syst. Appl.* **120**, 372–386 (2019).
45. Palmirani, M., Bincoletto, G., Leone, V., Sapienza, S. & Sovrano, F. Hybrid refining approach of pronto ontology. In *International Conference on Electronic Government and the Information Systems Perspective*, 3–17 (Springer, 2020).
46. Gatt, A. & Krahmer, E. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Int. Res.* **61**, 65–170 (2018).
47. Martínez Jiménez, J. A., Muñoz Marquina, F. & Sarrió Mora, M. Á. *Lengua Castellana y Literatura, 36* (Akal Sociedad Anónima, Madrid, España, Madrid, 2011).
48. Barraza Verdesoto, E. S., Rivas Trujillo, E. & Rodríguez Molano, J. I. Model texts with svo sentences as a system composed. structure such as the spanish language. *International Journal of Mechanical and Production Engineering Research and Development***10**, 16111–16118, https://doi.org/10.24247/ijmperdjun20201528 (2020).
49. Cohn, P. M. *Algebra*, vol. 3 (Jhon Wiley & Sons, 1991).
50. Miles, R. & Hamilton, K. *Learning UML 2.0* (O'Reilly, 2006).
51. Bick, E. A constraint grammar-based parser for spanish. In *TIL* (2006).
52. Barraza Verdesoto, E. S., Rivas Trujillo, E., Medina García, V. H. & Cardona Sánchez, D. Algebraic model to formalize sentences and their context: Use case scenario of the spanish language. In *Applied Computer Sciences in Engineering*, 182–193, https://doi.org/10.1007/978-3-030-00350-0_16 (2018).
53. Mann, W. C. & Thompson, S. A. Rhetorical structure theory: Toward a functional theory of text organization. *Text - Interdiscip. J. Study Discourse* **8**, 243–281. https://doi.org/10.1515/text.1.1988.8.3.243 (1988).
54. Mabona, A., Rimell, L., Clark, S. & Vlachos, A. Neural generative rhetorical structure parsing. arXiv preprint arXiv:1909.11049 (2019).

55. Hou, S., Zhang, S. & Fei, C. Rhetorical structure theory: A comprehensive review of theory, parsing methods and applications. *Expert Syst. Appl.* **157**, 113421. https://doi.org/10.1016/j.eswa.2020.113421 (2020).
56. Reiter, E. & Dale, R. *Building Natural Language Generation Systems* (Cambridge University Press, 2000).
57. Bouayad-Agha, N., Casamayor, G., Wanner, L. & Mellish, C. Overview of the first content selection challenge from open semantic web data. In *Proceedings of the 14th European Workshop on Natural Language Generation*, 98–102 (Association for Computational Linguistics, Sofia, Bulgaria, 2013).
58. Wanner, L. *et al.* Getting the environmental information across: from the web to the user. *Expert. Syst.* **32**, 405–432. https://doi.org/10.1111/exsy.12100 (2015).
59. Kutlak, R., Mellish, C. & van Deemter, K. Content selection challenge - University of Aberdeen entry. In *Proceedings of the 14th European Workshop on Natural Language Generation*, 208–209 (Association for Computational Linguistics, Sofia, Bulgaria, 2013).
60. Barzilay, R. & Lee, L. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, 113–120 (Association for Computational Linguistics, Boston, Massachusetts, USA, 2004).
61. Lapata, M. Automatic evaluation of information ordering: Kendall's tau. *Comput. Linguist.* **32**, 471–484. https://doi.org/10.1162/coli.2006.32.4.471 (2006).
62. Bollegala, D., Okazaki, N. & Ishizuka, M. A bottom-up approach to sentence ordering for multi-document summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 385–392, https://doi.org/10.3115/1220175.1220224 (Association for Computational Linguistics, Sydney, Australia, 2006).
63. Walker, M. A., Rambow, O. & Rogati, M. SPoT: A trainable sentence planner. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics* (2001).
64. Cheng, H. & Mellish, C. Capturing the interaction between aggregation and text planning in two generation systems. In *INLG'2000 Proceedings of the First International Conference on Natural Language Generation*, 186–193, https://doi.org/10.3115/1118253.1118279 (Association for Computational Linguistics, Mitzpe Ramon, Israel, 2000).
65. Kennedy, C. & McNally, L. Scale structure, degree modification, and the semantics of gradable predicates. *Language* 345–381 (2005).
66. Angeli, G., Liang, P. & Klein, D. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 502–512 (2010).
67. Kondadadi, R., Howald, B. & Schilder, F. A statistical nlg framework for aggregated planning and realization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1406–1415 (2013).
68. Gatt, A. & Krahmer, E. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Int. Res.* **61**, 65–170 (2018).

Author contributions

Edgardo Barraza is the main author, he generated the main idea of the paper and its proof previously published in another journal.Richard de JesÃºs is the doctoral tutor of Edgardo Barraza and he contributed to the direction of the research.Marlly Rojas wrote part of the paper and she was fundamental in final review

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to E.S.B.V.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.