Building natural language responses from natural language questions in the spatio-temporal context

Ghada Landoulsi*, Khaoula Mahmoudi and Sami Faïz

Laboratory of Remote Sensing and Information
Systems with Spatial References (LTSIRS),
ENIT National Engineering School of Tunis,
Tunis El Manar University,
El Manar 1, Tunisia
Email: ghadalandoulsi@yahoo.fr
Email: kamahmoudi@yahoo.fr
Email: sami.faiz@isamm.uma.tn
*Corresponding author

Abstract: With the evolving research in geographic information system (GIS) owing to its ability to support decision makers in different fields, there is a strong need to enabling all users; specialists and non-specialists to profit from this technology. Although, the key impediment to non-specialists is the language to interact with the GIS and especially its embedded geographic database (GDB) which require SQL skills. In this paper we explore a new approach which alleviates nomad GIS users from any formatting effort by only using the natural language as a GDB communication mean. The process is generally two-fold: 1) formatting the natural language user query to be processed by the GDB engine; 2) translating the GDB retrieved answer to a text easily interpreted by all GIS users. The resulting implemented system was integrated to the OpenJump GIS and has been evaluated to give satisfactory results.

Keywords: spatio-temporal data; geographic databases; GDBs; question answering systems; structured query language; natural language generation; NLG.

Reference to this paper should be made as follows: Landoulsi, G., Mahmoudi, K. and Faïz, S. (2021) ‘Building natural language responses from natural language questions in the spatio-temporal context’, *Int. J. Intelligent Information and Database Systems*, Vol. 14, No. 1, pp.1–25.

Biographical notes: Ghada Landoulsi is currently pursuing her PhD in Tunisia Polytechnic School (EPT). She is a member of Laboratory of Remote Sensing and Spatial Information Systems (LTSIRS) – ENIT. She is a holder of a Bachelor’s in Computer Management at the Faculty of Economics and Management of Nabeul in 2010, and Master’s of Data, Knowledge and Distributed Systems at the Faculty of Legal Sciences, Economy and Management of Jendouba in 2013. Her current research interests include geographic information systems and natural language processing.

Khaoula Mahmoudi is a holder of a PhD in Information and Communications Technology (ICT) at the Higher School of Communication of Tunis (SUP’COM). She obtained her Master’s in Business Data Processing at the Higher Institute of Management (ISG). She taught in several universities since

2001 and currently she is an Assistant Professor in Computer Science at the FSEGN – University of Carthage, Tunisia. She is a member of Laboratory of Remote Sensing and Spatial Information Systems – ENIT. She is concerned with research activities in the area of the enrichment of geographic databases in the context of geographic information systems (GIS). She presented and published numerous articles in different conferences books and journals.

Sami Faïz is actually a Full Professor in Computer Science at the Higher Institute of Multimedia Arts of Manouba and a member of the Laboratory of Remote Sensing and Spatial Information Systems (LTSIRS). Besides, he is the Director of the institute. He published more than 150 papers in specialised conferences and journals. He is also the author of four books in the framework of GIS. He is a scientific and organising member of various international conferences. He is also the founder of many national and international projects in Geomatics.

1 Introduction

The potential benefits of GIS is its ability to manage the information on both spatial and aspatial aspects relating to a decision problem. Hence, having easy access to the information relating to a decision problem is a key element in popularising the GIS. Indeed, to manage the dataset stored in the geographic database (GDB) the SQL skills are mandatory. To make this dataset accessible to the professionals and even to the general public we explore other communication means namely the natural language which is free from any formatted structure.

To this end, in this study we have exploited the question answering technique to minimise the communication gap between computers and non-technical users. Indeed, we do not refer to the SQL queries as commonly used input, instead we propose to launch through the GIS native interface queries in a natural language free from any formal structuring. In feedback, the usual systems produce different forms of answers (tables, maps, etc.). Again to assist the users to interpret the answers returned on a map or a table…, we recourse to response reformulation in a natural language fashion.

The remainder of this paper is organised as follows. Section 2 reviews related works. Section 3 describes our two staged approach. It deals with a first stage is dedicated to interpret and to translate the user natural language question into a machine understandable structure. This target is reached by generating a query with respect to the spatio-temporal structured query language (ST-SQL). Such query is used to extract the desired information from a given GDB. Then a second stage involves reformulating the extracted answers to finally generate a more semantic textual description of them. Section 4 details our simulation results. Section 5 presents our evaluation process. Finally, the last section reports the conclusion with some perspectives.

2 Related works

With the information glut, it is of prime importance to provide means to experts as well as novice users to easily access to data repositories and get relevant answers. One way is to establish facilities to ensure the communication with the machine and the end users through the natural language natively used by humans. Hence without leaning on the formal language a set of techniques was designed to reach this target, namely question answering systems (QAS), the natural language generation (NLG), etc. The latter are the subjects of a literature surveys since they are the scope of the main stages of the current study.

2.1 Question answering systems

Question answering systems broadly fall into two basic domains: general domain and restricted domain. For the first category, questions are addressed by the wide range of users. Regarding restricted domain QASs, they are more suitable to expert users concerned with asking questions for specific domains. Indeed, the literature review has revealed that QASs concerned with geographic information have not acquired a lot of attention and only few works have addressed such information. In what follows, we devote the following review to the general and geographical domain related works.

In Green et al. (1961) authors proposed BASEBALL, a QAS that retrieves information related to the baseball league played in America. The answers concern dates, locations, etc. In the same context, Woods (1977) proposed LUNAR a QAS that retrieves information about soil samples provided from Apollo lunar exploration. The main objective of BASEBALL and LUNAR systems is to transform questions into database queries using pattern matching rules. These latter, use limited grammars, hard wired knowledge, and mapping rules. In spite of their effectiveness, these systems have a limited repository of information related to their application domains. NOMAO (Delpech and Candillier, 2012) represents another GQAS specialised in places recommendation and e-reputation. The information is retrieved from websites and places directory to answer questions about types of locations and geographical areas. Recently, NaLIR (Li and Jagadish, 2014) is another system which uses dependency parse trees and several rules and heuristics to generate SQL queries. The SQL generation process is based on building candidate query trees which is considered as an intermediate step. A scoring mechanism is used to compute the appropriate query tree. This mechanism is based on computing the similarity between the dependency and the query trees and also between the adjacent nodes in a query tree. Yaghmazadeh et al. (2017) proposed Sqlizer, a system which mixes between rule-based and heuristics approaches. The authors generate a query sketch using a semantic parser. In fact, it represents the structure of an SQL query, including the different clauses and statement without any specific database schema information. Rules and heuristics are proposed and used to refine and repair the query sketch to generate a candidate SQL query.

By taking inspiration from the different methods and techniques proposed in these studies, and in order to accomplish the first stage of our approach, we propose in this paper a heuristics-based approach. This latter is used to improve search performance of the QASs by considering the spatio-temporal aspect in the user questions.

2.2 NLG systems

Nowadays, the revolution of the real-life applications has an increasing impact in the NLG approaches which try to find successful solutions for texts generation problems. In this context, the main quite research issue presented within NLG is to produce texts which are comprehensible by casual users and that fulfil their needs. In what follows we are focusing on the most relevant data-to-text NLG systems which have been proposed affording a practical use.

The meteorology domain represents one of the domains in which a variety of data to text systems have been deployed to produce weather forecasts reports. FoG (Goldberg et al., 1994) is one of the pioneers in this context. This latter, is dedicated to generate marine weather forecasts reports using several rules and formal grammars. The medical domain is considered as a very interesting field where the NLG technology has been deployed where several systems are proposed to summarise clinical data. TOPAZ is a decision-support system (Kahn et al., 1991) which generates summaries from discrete information such as blood cell counts and drug dosages of lymphoma patients' data. The proposed approach is dedicated to detect deviations where the TOPAZ system (Kahn et al., 1991) compared patient specific values to population parameters utilising a numerical model. Furthermore, it is dedicated to join events together forming temporal abstractions with possible explanations. These abstractions are then converted into a textual summary that could be used by clinicians. In the same context, the baby talk (BT) family of systems (Hunter et al., 2012), are considered as the first medical NLG systems that produce summaries for sensor and discrete information. These systems are mainly based on various techniques such as knowledge engineering, signal processing, etc.

The data-to-text generation process can be performed by applying three main steps (Danlos, 1989). First of all, a document planning determines the final form of the textual descriptions to be generated. Then, a surface planning transforms the document plan into a sequence of sentences by choosing the words that will express them. Finally, a surface realisation translates the conceptual representation into text. In early years, many studies were concentrated on surface realisation step.

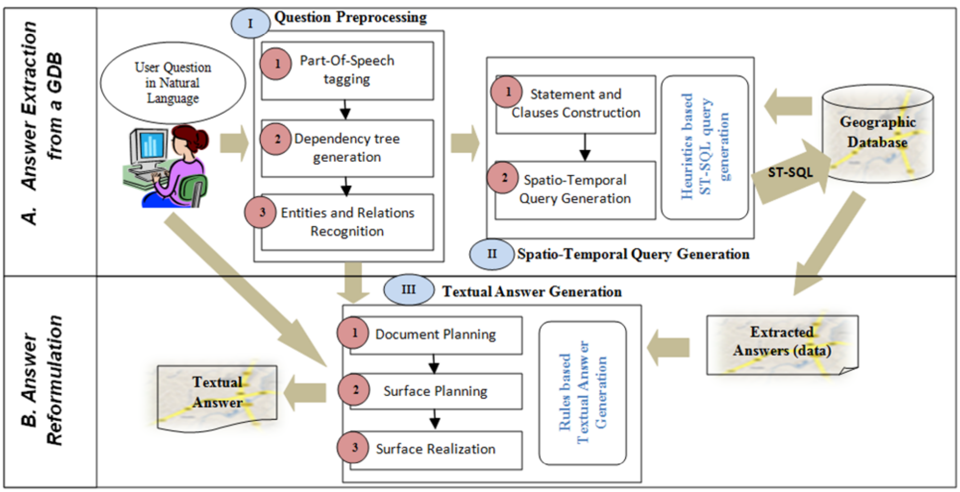
Researchers apply template-based approaches (Van Deemter et al., 2005) and employ several shallow models such as: probabilistic context-free grammars (Belz, 2008) and rule-based models (Angeli et al., 2010). Concerning the document and surface planning steps, researchers propose machine learning approaches based on a generative semi-Markov model (Liang et al., 2009) or consider the planning steps as a collective classification problem (Barzilay and Lapata, 2005).

Nowadays, recurrent neural networks (RNNs) play a significant role in data-to-text generation. RNNs are used in a variety of applications dealing with the machine translation process (Bahdanau et al., 2015), automatic summarisation process (Tan et al., 2017), etc. For the data-to-text generation process several approaches are proposed using RNNs. As an example, a coarse-to-fine grained attention mechanism is proposed by Mei et al. (2016) to pick out records by using a probability and then to capture the relevant content. In the same context, Lebret et al. (2016) integrate the copy mechanism into the data-to-text generation process. By studying the above approaches, we inferred that researchers do not explicitly deal with the content structuring during the document planning step. Nevertheless, determining the order of the content according to a logical structure is a very interesting task to ensure a generation of a well-written text. Our approach focuses on the document planning step and more specifically the content structuring sub-step.

3 Proposed approach

The approach emanates from the need to democratise the GIS exploitation to all kind of users. Hence experts and non-experts will take benefit from a natural workflow to retrieve the required data. The approach we propose relies on the idea that the communication between GIS and end users is ensured by adopting only the natural language. Indeed, a user looking for some information has to formulate a natural question. The latter is to be answered by the GDB engine. Hence an SQL generation process is performed to return the appropriate information. The latter will be taken as the input in the subsequent stage to generate the equivalent answered text to be provided to the end user (see Figure 1). In what follows we detail the different steps in turn.

Figure 1 System architecture (see online version for colours)



3.1 Answer extraction from a GDB

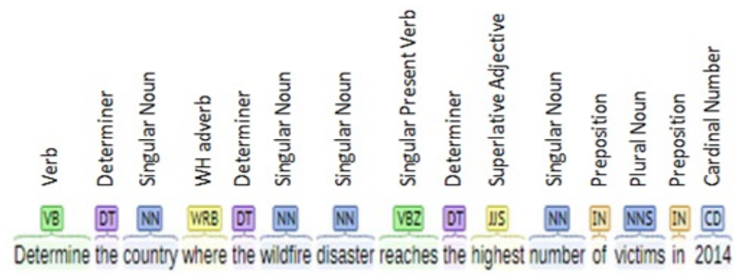
Computers are unable to interpret and to understand natural language questions used by human for communication. To this end natural language processing (NLP) technique plays a vital role to interpret the natural language question correctly and to translate it into a structured format. The system starting point is to submit an input in a natural language format in order to get the required information from a GDB.

To make the GDB data retrieving possible a formal question structuring is mandatory. To this end the system has to execute the following pipelined steps, namely: a question pre-processing step and then a spatio-temporal SQL (ST-SQL) query generation from a natural language question one.

3.1.1 Question pre-processing

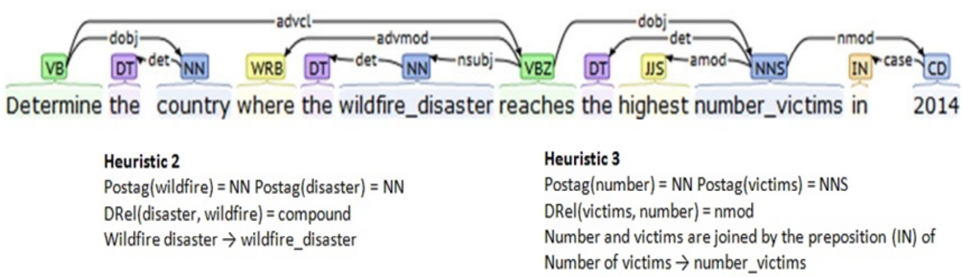
This step is executed to make the interpretation of the user question by the machine a straightforward task. The pre-processing step consists of three main sub-steps: POS-tagging, dependency tree generation and spatio-temporal entities and relations recognition. The user question must be first tagged by a part-of-speech tagger (POS tagger). This latter is used to assign the grammatical category to each question term. For the sake of illustration, we give the following spatio-temporal user question example ST_{uq} (ex) to be pre-processed (see Figure 2): "determine the country where the wildfire disaster reaches the highest number of victims in 2014."

Figure 2 Part-of-speech tagging of the ST_{uq} (ex) (see online version for colours)



Thus, the different lexical relations existing between each pair of words in a ST_{uq} must be identified using a dependency parser (Marneffe et al., 2006). This latter, is dedicated to assign the grammatical relation between word (W_i) and word (W_j) (see Figure 3).

Figure 3 Dependency relations representation for the ST_{uq} (ex) (see online version for colours)



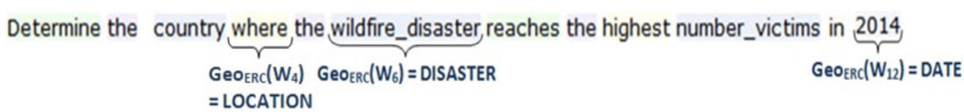
In this paper, we have chosen natural disasters as the field of our study. In general, a spatio-temporal natural language query involves searches in the spatial and temporal dimensions. This query type is dedicated to describe events by identifying their location name, location type, date, duration, temporal relation and spatio-temporal relation.

In the literature, four types of spatio-temporal queries are mentioned (Yuan and McIntosh, 2002). Simple spatio-temporal query, spatio-temporal range query, spatio-temporal behaviour query and spatio-temporal relationship query. In the present paper we deal only with two classes of queries: the simple spatio-temporal query and the spatio-temporal range one. The first class allows seeking for information about the location of an object at a given time, the time when an object can be detected at a particular location or the object that can be spotted at a particular location and at a given time. Regarding the spatio-temporal range query type, it looks for information about the changes that may occur in a region during a given time period.

In natural language questions, users can use ambiguous expressions to describe their information requirements. To resolve such ambiguous situations, we use a named entity recogniser (Manning et al., 2014) to identify entities and relations that the query may contain. In the context of natural disasters all the geo-coordinates are labelled with LOCATION class such as forest, lands, country, etc. In the same context, all types of disasters such as wildfire, explosion and hazardous materials... are identified as DISASTER class. Regarding the temporal expressions we adopted a rule-based temporal tagger (Chang and Manning, 2012) which supports five types of temporal expressions: DATE, TIME, DURATION, SET and INTERVAL.

In past studies only spatial (equals, disjoint, intersects, etc.) (Worboys, 1992), or temporal relations (during, started, finished, etc.) (Allen, 1984) are considered. While valid spatial relations for a particular period of time are categorised as spatio-temporal relations (Raza, 2008). These latter are identified using set of regular expressions (Chang and Manning, 2014) able to detect each type of relation. Furthermore, if a user introduces a WH question, the corresponding words such as (where, when, how longer, etc.) should be identified. These words reveal implicitly what the user is looking for. For example, the WH question word 'where' can be labelled with LOCATION class and 'How_longer' can be labelled with DURATION class. In Figure 4 we illustrate the geographic entities recognition depicted within the $ST_{uq(ex)}$.

Figure 4 Entities recognition example (see online version for colours)



3.1.2 Spatio-temporal query generation from natural language question

Our approach to generate ST-SQL queries is reached using the simple and the expressive formulation of the spatio-temporal queries. The formulation that we adopt preserves the general syntax of the classical SQL queries with the integration of functions and aggregates related to spatial, temporal and spatio-temporal aspects.

This formulation respects the following syntax: SELECT attribute(s)_name(s) FROM table(s)_name(s) WHERE condition(s) GROUP BY attribute(s)_name(s) HAVING condition(s). In what follows, we showcase the different steps leading to generate ST-SQL query components from a natural language question. The SELECT statement construction process is achieved using a set of heuristics (see Figure 5). These latter rely upon the list of the part-of-speech tags, the list of the dependency relations and the recognised geographic entities and relations [see Figure 10 – (1)].

In association with the SELECT statement (Sst) it is common to use the WHERE clause (Wc) to retain only data which meet the criteria specified in the ST_{uq}. In such clause a set of heuristics are determined by referring to the POS tags and the geographic entities and relations classes (denoted Geo_{ERC}) to identify the predicates [see Figure 6 and Figure 10 – (2)].

Figure 5 SELECT statement construction

Algorithm 1

```
Input STuq = {w1, w2, w3…, wn}, WHC
// WHC = WH question words Classes
Output SstattL
Begin
SstattL←∅
For each word W ∈ STuq do
//Heuristic 1
If (POStag(Wi) ∈ {NN, NNS}) and (DRel(Wj, Wi) ∈  {root, nsubj, dep, dobj} Then
    Add Wi to SstattL
End if
//Heuristic 2
If (POStag(Wi) ∈ {NN, NNS}) and (POStag(Wj) ∈ {NN, NNS}) and (DRel(Wj, Wi) ∈
{conj:and})) Then // conj:and = Conjunction
     Add Wi and Wj to SstattL
 End if
//Heuristic 3
  If (POStag(Wi) ∈ {NN, NNS}) and (POStag(Wj) ∈ {NNP}) and (DRel(Wj, Wi) ∈
{nmod, compound}) Then
  Add Wi to SstattL
End if
//Heuristic 4
If (POStag(Wi) ∈ {NN, NNS}) and (POStag(Wj) ∈ {NN, NNS}) and (DRel(Wj, Wi) ∈
{nmod}) Then
  Add Wi and Wj to SstattL
End if
//Heuristic 5
 If (POStag(Wi) ∈ {WRB, WP}) Then //WRB = WH adverb,
       WP = WH pronoun
Add the WHC of Wi to SstattL
End if
End do
End
```

Figure 6 WHERE clause construction

Algorithm 2

```
Input STuq = {w1, w2, w3…, wn}, GeoERC
Output Wcconditions // WHERE clause conditions list
Begin
Wcconditions←∅
For each word W ∈ STuq do
//Heuristic 1
If (POStag(Wi) ∈ {NNP, NNPS, CD}) Then
    Wcconditions ← (GeoERC(Wi) = 'Wi')
//Heuristic 2
Else if (POStag(Wi) ∈ {NN, NNS}) followed by (POStag(Wi+1) ∈ {VBZ}) followed by
(POStag(Wi+2) ∈ {NNP, NNPS}) Then
    Wcconditions ← (Wi = 'Wi+2')
//Heuristic 3
Else if (POStag(Wi) ∈ {NN, NNS}) and (GeoERC(Wi) = value) Then
    Wcconditions ← (GeoERC(Wi) = 'Wi')
//Heuristic 4
Else if (POStag(Wi) ∈ {NN, NNS, NNP, NNPS}) and (POStag(Wi+1) ∈ {JJR}) Then
    Calculate the similarity between words: (Wi+1), Inferior and Superior
    If Similarity (Wi+1,"Inferior") ≥ threshold Then
        Wcconditions ← Wi < 'Wi+3'
    Else if Similarity (Wi+1, "Superior") ≥ threshold Then
        Wcconditions ← Wi > 'Wi+3'
    End if
End for
End
```

Figure 7 HAVING clause construction (heuristic 1)

Algorithm 3

```
Input ST_uq = {w_1, w_2, w_3…,w_n }, SA_op, TA_op, STA_op
Output Hc_conditions
Begin
For each word W ∈ ST_uq do
  If (POStag(W_i) ∈ {NN, NNS, NNP, NNPS}) and followed by (POStag(W_j) ∈ {JJR}
Then
    For each (SA_op) do
      If the Similarity (W_i, (SA_op(k))) ≥ threshold Then
         SA_att = SA_op(k)(W_i) //SA_att = spatial aggregate attribute
         If the similarity (W_j, "Inferior") ≥ threshold Then
           Hc_conditions ← SA_op(k)(W_i) < W_j+2
         Else if the similarity (W_j, "Superior") ≥ threshold Then
           Hc_conditions ← SA_op(k)(W_i) > W_j+2
         End if
       End if
     End for
     For each (TA_op) do
       If the Similarity (W_i, (TA_op(l))) ≥ threshold Then
         TA_att = TA_op(l)(W_i) //TA_att: temporal aggregate attribute
         If the Similarity (W_j, "Inferior") ≥ threshold Then
           Hc_conditions ← TA_op(l)(W_i) < W_j+2
         Else if the similarity (W_j, "Superior") ≥ threshold Then
           Hc_conditions ← TA_op(l)(W_i) > W_i+2
         End if
       End if
    End for
    For each (STA_op) do
       If the Similarity (W_i, (STA_op(m))) ≥ threshold Then
          STA_att = STA_op(m)(W_i) //STA_att: spatio-temporal aggregate attribute
          If the Similarity (W_j, "Inferior") ≥ threshold Then
             Hc_conditions ← STA_op(m)(W_i) < W_j+2
          Else if the Similarity (W_j, "Superior") ≥ threshold Then
             Hc_conditions ← STA_op(m)(W_i) > W_i+2
          End if
       End if
     End for
   End if
 End for
End
```

In this algorithm, the first three heuristics consist of checking if a word i corresponds to a well-defined grammatical category, then the WHERE clause represents an equality between the class of an entity (word) i and the word i (Geo_{ERC} (W_i) = 'W_i'), or between two words (W_i = 'W_{i+2}'). Concerning heuristic 4, we assume that whether the word W_i corresponds to one of the following grammatical categories {NN, NNS, NNP, NNPS} and it is followed by a comparative adjective (W_{i+1}), the semantic similarity between the adjective (JJR) (W_{i+1}) and the words 'inferior' and 'superior' is computed to depict the WHERE clause condition(s). If the similarity between W_{i+1} and the word 'inferior' is greater than a given threshold then the noun (NN or NNS) or the proper noun (NNP or NNPS) W_i is less than the value W_{i+3}. W_{i+2} must be necessarily a preposition which connects the adjective W_{i+1} and the value W_{i+3} ($Wc_{conditions} \leftarrow W_i <$ 'W_{i+3}'). Otherwise, if the similarity between W_{i+1} and the word 'superior' is greater than a given threshold then W_i is greater than the value W_{i+3} ($Wc_{conditions} \leftarrow W_i >$ 'W_{i+3}').

Figure 8 HAVING clause construction (heuristic 2)

Algorithm 4

```
Input ST_UQ = {w_1,w_2,w_3…,w_n}, SA_op, TA_op, STA_op, Wc_conditions, Fc
Output Hc_conditions
Begin
 For each word W ∈ ST_uq do
  If (POStag(W_i) ∈ {JJS}) followed by (POStag(W_j) ∈ {NN, NNS})
    If the Similarity (W_i, "Inferior") ≥ threshold Then
      Hc_conditions ← W_j = Select MIN(W_j From Fc Where Wc_conditions)
    Else if the Similarity (W_i, "Superior") ≥ threshold Then
      Hc_conditions ← W_j = Select MAX(W_j From Fc Where Wc_conditions)
    Else
      For each (SA_op) do
        If the Similarity (W_j, (SA_op(k))) ≥ threshold Then
           SA_att = SA_op(k)(W_j)
           If the Similarity (W_i, "Inferior") ≥ threshold Then
              Hc_conditions ← SA_att = Select MIN(SA_att From Fc Where Wc_conditions)
           Else if the Similarity (W_i, "Superior") ≥ threshold Then
              Hc_conditions ← SA_att = Select MAX(SA_att From Fc Where Wc_conditions)
           End if
        End if
      End for
      For each (TA_op) do
        If the Similarity (W_j, (TA_op(l))) ≥ threshold Then
           TA_att = TA_op(l)(W_j)
           If the Similarity (W_i, "Inferior") ≥ threshold Then
              Hc_conditions ← TA_att = Select MIN(TA_att From Fc Where Wc_conditions)
           Else if the Similarity (W_i, "Superior") ≥ threshold Then
              Hc_conditions ← TA_att =Select MAX(TA_att From Fc Where Wc_conditions)
            End if
          End if
        End for
        For (STA_op) do
          If the Similarity (W_j, (STA_op(m))) ≥ threshold Then
             STA_att = STA_op(m)(W_j)
             If the Similarity (W_i, "Inferior") ≥ threshold Then
               Hc_conditions ← STA_att=Select MIN(STA_att From Fc Where Wc_conditions)
             Else if the Similarity (W_i, "Superior") ≥ threshold Then
               Hc_conditions ← STA_att=Select MAX(STA_att From Fc Where Wc_conditions)
             End if
           End if
         End for
       End if
     End for
End
```

Various aggregates can be applied supporting spatial aggregate operators (SA_{op}): area, perimeter, distance, etc. Temporal aggregate operators (TA_{op}) of Allen (1984): duration, contain, intersect, etc. Finally, spatio-temporal aggregate operators (STA_{op}) like moving-distance. The HAVING clause (Hc) can be used with these different aggregate operators to retrieve summary values for the attributes given in the GROUP BY clause [see Figure 10 – (3)]. The construction process of the Hc is mainly based on two heuristics. These latter are dedicated to identify the aggregate attributes encapsulated with comparative or superlative adjectives. The first heuristic focuses on the idea that whether W_i is a noun or a proper noun and followed by a comparative adjective (denoted JJR) the aggregate attributes (SA_{att}, TA_{att}, STA_{att}) should be identified. In this case we adopt the

Wu-Palmer semantic similarity measure (Wu and Palmer, 1994) to calculate the similarity between W_i and the aggregate operators (SA_{op}, TA_{op}, STA_{op}). The following algorithms describe the two heuristics dedicated to determine the Hc conditions ($Hc_{conditions}$) list.

The second heuristic in Figure 8 focuses on the idea that if W_i is a superlative adjective (denoted JJS) followed by a noun it is required to use the aggregate functions MAX () and MIN () in a sub query.

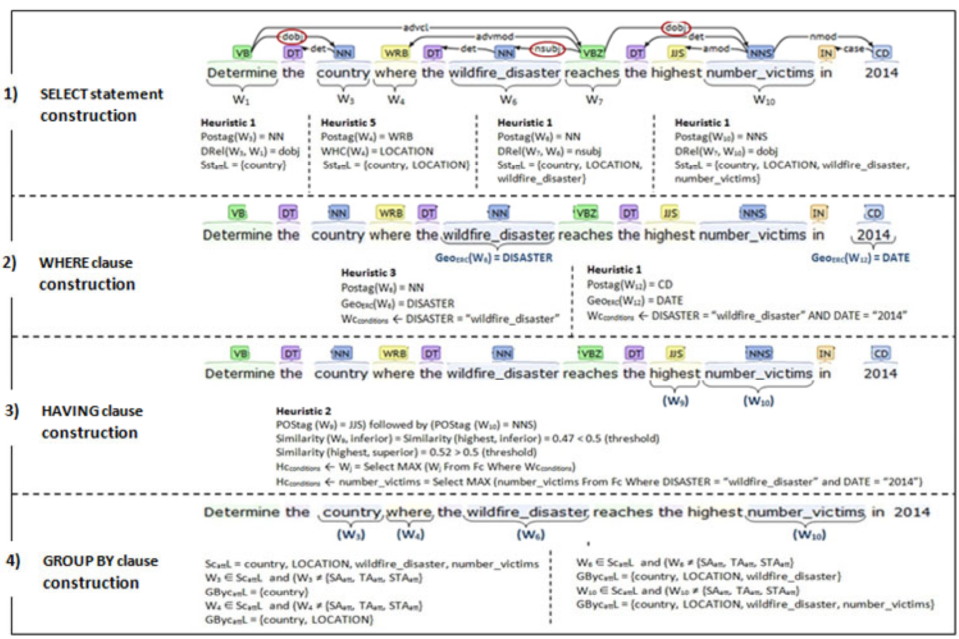
The Sst is most often combined with the GROUP BY clause (GByc). This latter includes attributes that are not encapsulated within a spatial, temporal or spatio-temporal operator [see Figure 9 and Figure 10 – (4)].

Figure 9 GROUP BY clause construction

Algorithm 5

```
Input Sc_attL
Output GByc_attL
GByc_attL←∅
Begin
   For each word W ∈ ST_uq do
     If (W_i ∈ Sc_attL) and (W_i ∉{SA_att, TA_att, STA_att}) Then
       Add W_i to GByc_attL
     End if
   End for
End
```

Figure 10 Statement and clauses construction example (see online version for colours)



Reaching this level, the SELECT statement and the different clauses are depicted without referring to the database table(s). Yet, these latter are compulsory part of the ST-SQL query. To refer to such tables we must use the FROM clause (denoted Fc). For this aim, we suggest to realise correspondence between the attributes of the constructed statement and clauses and the attributes of a GDB. To ensure this correspondence we adopt the Wu-Palmer similarity/relatedness measure (Wu and Palmer, 1994). This latter is used to compute either the semantic similarity or the semantic relatedness between attributes by taking into account their path length in the WordNet taxonomy (George, 1995). This measure is mainly based on the least common subsumer (LCS) which depicts the most specific common ancestor of each pair of words:

$$\mathrm{Sim}\left(\mathrm{word}_1, \mathrm{word}_2\right) = \frac{2\mathrm{depth}\left(\mathrm{LCS}\left(\mathrm{word}_1, \mathrm{word}_2\right)\right)}{\mathrm{depth}\left(\mathrm{word}_1\right) + \mathrm{depth}\left(\mathrm{word}_2\right)} \tag{1}$$

In the aforementioned equation the depth(LCS(word_1, word_2)) depicts the number of IS-A relationships from the most common word to the root of the taxonomy. depth(word_1) and depth(word_2) depict the number of IS-A relationships from word_1 and word_2 to the root (Guessoum et al., 2016). In the following algorithm we describe our method to build the FROM clause (Fc).

Figure 11 FROM clause construction

Algorithm 6

```
Input Sst_attL, Wc_attL, Hc_attL, GDB
Output Fc_attL // FROM clause attributes list
Fc_attL ←∅
Begin
   For each Table ∈ GDB do
     Compute the semantic similarity between the attributes of the Table_i and
   the attributes (Sst_attL, Wc_attL, Hc_attL).
     If the Similarity ≥ threshold then
        Add Table_i to Fc_attL
        Substitute the attributes (Sst_attL, Wc_attL, Hc_attL) with the Table_i
   attributes
     End if
   End for
End
```

Once more than one table is interrogated, the Fc could be a join operation which gathers multiple tables. In some cases, the database tables may contain ambiguous columns names. So, it is difficult to find a semantic similarity or even a semantic relatedness between the generated query attributes and the database tables attributes. In order to overcome this problem, we propose in the following algorithm to ensure correspondence by measuring the semantic relatedness between the attributes of the (Sst, Wc, GByc, Hc) and the values that each attribute of the database tables can take.

3.2 *Answer reformulation*

Since this study is dedicated to users who are more familiar with natural language questions than the formal one, the answer has to reflect the same target. Hence, once the response is retrieved from the GDB it has to be reformulated in an expressive natural language representation to be understandable by the large communities of users.

Figure 12 Correspondence between the ST SQL attributes and values that each attributes of the GDB tables can take

Algorithm 7

```
Input SstattL, FcattL, WcattL, GBycattL, HcattL, V1Atti // V1Atti: first value
associated to each Tablei attribute.
Output ST-SQL
Begin
  For each Table ∈ GDB do
    Compute the semantic-relatedness between the attributes (SstattL, WcattL,
  GBycattL, HcattL) and V1Atti
    If semantic-relatedness ≥ threshold Then
      Substitute the attributes (SstattL, WcattL, GBycattL, HcattL) with the
  appropriate attributes of the Tablei
      Formulate the ST-SQL query using (SstattL, FcattL, WcattL, GBycattL, HcattL)
    End if
End for
End
```

Our solution is mainly based on the data to text method. The basic idea is to use the ST_{uq} and the returned answers as inputs. The process of generating the natural language answer consists of three main steps (Danlos, 1989): document planning, surface planning and surface realisation. Each of these steps is described in detail below.

3.2.1 Document planning

In order to determine the content of the textual answer to be generated we propose to identify first the geographic feature in the ST_{uq} which will be the focus of the textual answer to be generated. In our case the focus feature is the recognised geographic entity. Second, to generate our textual answer we propose to use the ST_{uq} and the extracted answers as inputs. Once the content is determined, it must be structured.

The structuring of the content consists of identifying the textual template which can be then translated into linguistic expressions. This function takes the user question (ST_{uq}), the geographic entities and relations classes (Geo_{ERC}), the list of part-of-speech tags (POS tags) and the extracted answers as inputs. As output, our method transforms the interrogative form of the ST_{uq} into an affirmative form to represent the template of our textual answer.

In this context, the templates will define different patterns and they will be instantiated according to a set of proposed rules. These rules are used to provide flexibility in generating the textual answer, taking into account the ST_{uq} type, the extracted answers ($E_{Answers}$) (single or multiple data) and especially the geographic feature (Geo_f). In fact, we depict the following cases.

Case 1 If the focus of the ST_{uq} is a geographic feature and the $E_{Answers}$ is a single data then the rule will have the following format:

$$QwordType(W_{i=1},POStag)\wedge C_1 \wedge \ldots \wedge C_n \rightarrow Answer_{Template}(ST_{uq}, [x, \text{“is”}, data_{value}])$$

Case 2 If the focus of the ST_{uq} is a geographic feature and the $E_{Answers}$ are multiple data then the rule will have the following format:

$QwordType(W_{i=1},POStag)\wedge C_1\wedge\ldots\wedge C_n \rightarrow Answer_{Template}(ST_{uq}$, [x, "are", $data_{value(1)}$, "and", $data_{value(2)}$, "and",… $data_{value(n)}$])

In general, these rules include the predicate $QwordType(W_{i=1}, POStag)$. This latter indicates that the type of the question $word_{i=1}$ is the POStag of $W_{i=1}$. Thus, a set of conditions C_n are used. As an example, we use the predicate $Geo_{ERC}(WH, y)$ which indicates that the class name (Geo_{ERC}) of the WH question word is y.

On the other hand, the proposed rules include the predicate $Answer_{Template}(ST_{uq}, y)$ which means that the textual answer of the spatio-temporal user question is y. This latter, can be represented by a set of text strings concatenated with the $Answer_{Template}$ of the ST_{uq} (x). The different rules that satisfy the two cases are described in the following algorithm.

Figure 13 Document planning (case 1)

Algorithm 8

```
Input STuq, GeoERC, EAnswers // EAnswers: Extracted Answers
Output AnswerTemplate
Begin
  For each word W ∈ STuq do //Content Determination
  Identify the Geographic Entity (GeoE)
    If (Wi = GeoE) Then
       Geof = GeoE
    End if
// Content Structuring 1st Case
If (EAnswers = datavalue) and (Wi = Geof) Then
//Rule 1
  If QwordType(Wi-1) = verbe Then
     AnswerTemplate(STuq)= Concatenation["Wi+1…Wn", "is", datavalue]
//Rule 2
  Else If (QwordType(Wi-1) = WH) and (GeoERC(WH) = NUMBER) and (QwordType(Wi+1)
  = (NOUN) followed by (verb to be)) Then
     AnswerTemplate(STuq) = Concatenation["NUMBER", "of", Noun, "that", to be,
"Wi+3…Wn", "is", datavalue]
//Rule 3
  Else If (QwordType(Wi-1) = WH) and (GeoERC(WH) = NUMBER) and (QwordType(Wi+1) =
(NOUN) followed by (verb to do)) Then
     AnswerTemplate(STuq) = Concatenation["NUMBER", "of", Noun, "that", "Wi+3…Wn",
"is", datavalue]
//Rule 4
  Else If (QwordType(Wi-1) = WH) and (GeoERC(WH) = DURATION) and
  (QwordType(Wi+1) = (verb to do) followed by (NOUN) followed by (verb)) Then
     AnswerTemplate(STuq) = Concatenation [Noun, verb, "for", datavalue]
// Rule 5
  Else If (QwordType(Wi-1) = WH) and (GeoERC(WH) = DISTANCE) and (QwordType(Wi+1)
= (verb to be) followed by (adjective) followed by (Noun)) Then
     AnswerTemplate(STuq) = Concatenation["The", adjective, Noun, verb, "for",
datavalue]
//Rule 6
  Else If (QwordType(Wi-1) = WH) and (GeoERC(WH) = LOCATION) and
  (QwordType(Wi+1) = (verb to be) followed by (Noun) fo byllowed (verb)) Then
     AnswerTemplate(STuq) = Concatenation[Noun, verb, "for", datavalue]
//Rule 7
  Else If (QwordType(Wi-1) = WH) and GeoERC(WH) = OBJECT) and (QwordType(Wi+1) =
  (verb to be="is")) Then
     AnswerTemplate(STuq) = Concatenation["Wi+2…Wn", "is", datavalue]
  End If
  End If
  End
```

Figure 14 Document planning (case 2)

Algorithm 9

```
Input ST_uq, Geo_ERC, E_Answers // E_Answers: Extracted Answers
Output Answer_Template
Begin
  For each word W ∈ ST_uq do //Content Determination
  Identify the Geographic Entity (Geo_E)
    If (W_i = Geo_E) Then
       Geo_f = Geo_E
    End if
// Content Structuring 2nd Case
 If (E_Answers = data_value(1), data_value(2), data_value(n)) et (W_i = Geo_f) Then
//Rule 1
  If (QwordType(W_i=1) = (verb) followed by (Noun(NNS))) Then
  Answer_Template(ST_uq) = Concatenation["W_i+1…W_n", "are", ", data_value(1), "and",
  data_value(2), "and",… data_value(n)]
//Rule 2
  Else If (QwordType(W_i=1) = WH) and Geo_ERC(WH) = OBJECT) and (QwordType(W_i+1) =
  (verb to be="are")) Then
    Answer_Template(ST_uq) = Concatenation["W_i+2…W_n", "are", data_value(1), "and",
  data_value(2), "and",… data_value(n)]
  End If
  End If
End
```

Figure 15 Document planning schemas (case 1)

- **Schema 1 for Rule 1**: «Concatenation ["$W_{i+1}\dots W_n$", to be (singular), $data_{value}$]» :
  S → NP VP NP
  NP → "$W_{i+1}\dots W_n$"
  "$W_{i+1}\dots W_n$" → Det Noun PP
  PP → AdjP prep Noun prep Number
  PP → prep $data_{value}$
  AdjP → Det Adjective Noun
  VP → to be (singular)
  NP → $data_{value}$
  $data_{value}$ → Name (ex : country)
  $data_{value}$ → Number (ex : year, distance, area, etc) Adjective
  $data_{value}$ → Number (ex : year, distance, area, etc) Noun
  $data_{value}$ → Det Noun
  $data_{value}$ → Det Noun Name

- **Schema 2 for Rule 2 and Rule 3:** «Concatenation ["NUMBER", "of", Noun, "that", $verb_{i+3}$ (past tense) (passive set), "$W_{i+4}\dots W_n$", to be (singular), $data_{value}$] » :
  S → NP VP NP
  NP → "NUMBER", "of", Noun, "that", $verb_{i+3}$ (past tense) (passive set), "$W_{i+4}\dots W_n$"
  "$W_{i+4}\dots W_n$" → PP
  PP → prep NP
  PP → AdjP prep Noun
  AdjP → Det Adjective Noun
  NP → Name (ex : country)
  NP → Number (ex : year, distance, area, etc)
  NP → Det Noun
  NP → Det Noun Name
  VP → Verb to be (singular)
  NP → $data_{value}$
  $data_{value}$ → Number

- **Schema 3 for Rule 4:** «Concatenation [Noun, verb, "for", $data_{value}$] » :
  s → NP VP NP
  NP → Det Noun
  VP → Verb prep ("for")
  NP → $data_{value}$
  $data_{value}$ → Duration

- **Schema 4 for Rule 5** «Concatenation ["The", adjective, Noun, verb, "for", $data_{value}$] » :
  s → NP VP NP
  NP → adjective Noun
  VP → Verb prep ("for")
  NP → $data_{value}$
  $data_{value}$ → Distance

- **Schema 5 for Rule 6:** «Concatenation [Noun, verb, spatial relation, $data_{value}$] » :
  S → NP VP NP
  NP → Noun
  VP → verb PP
  VP → topological verbs (Equals, Disjoint, Intersects, Touches, Contains, etc)
  PP → spatial relation NP
  spatial relation → Directional relations (left, on the back, athwart, on the right of, behind, in front of)
  spatial relation → Distance relations (at, nearby, in the vicinity, far away)
  NP → $data_{value}$
  $data_{value}$ → Location

- **Schema 6 for Rule 7:** «Concatenation ["$W_{i+2}\dots W_n$", to be, $data_{value}$] » :
  S → NP VP NP
  NP → "$W_{i+2}\dots W_n$"
  "$W_{i+2}\dots W_n$" → Det Noun PP
  PP → prep NP
  PP → AdjP prep Noun prep Number
  AdjP → Det Adjective Noun
  VP → Verb to be (singular)
  NP → $data_{value}$
  $data_{value}$ → Name (ex : country)
  $data_{value}$ → Number (ex : year, distance, area, etc)
  $data_{value}$ → Det Noun
  $data_{value}$ → Det Noun Name

The answers templates (denoted $Answer_{Template}$) play an organising role to decide how these pieces of information can be related to each other and in what order they should appear. To ensure this structuring, we propose the use of schemas (Pereira and Warren, 1980), which specify how a document plan can be constructed. A schema is an element sequence grammar. This grammar can be described in definite clause grammar (DCG) notation. The schemas in Figures 15 and 16 can be deduced from the answer templates which correspond to each rule defined in Algorithms 8 and 9.

s → NP VP NP represents a sentence with NP is a nominal phrase, VP is a verbal phrase, PP is a prepositional phrase, and AdjP is an adjectival phrase. The first NP of the sentence 's' represents the segment 'W_{i+1} ... W_n' which can be represented by the concatenation of a determinate (Det), with a noun and a PP.

Figure 16 Document planning schemas (case 2)

- **Schema 7 for Rule 1:** «Concatenation ["W_{i+1}…W_n"; Verb to be (plural), $data_{value(1)}$, "and", $data_{value(2)}$, "and",… $data_{value(n)}$] » :
  s → NP VP NP
  NP → "W_{i+1} …W_n"
  "W_{i+1} …W_n" → Det Noun PP
  PP → prep NP
  PP → AdjP prep Noun prep Number
  AdjP → Det Adjective Noun
  VP → verb to be (plural)
  NP → $data_{value(1)}$, "and", …$data_{value(n)}$
  $data_{value(1)}$, "and", … $data_{value(n)}$ → $Name_1$, "and", …$Name_n$
  $data_{value(1)}$, "and", … $data_{value(n)}$ → $Number_1$, "and", …$Number_n$
  $data_{value(1)}$, "and", … $data_{value(n)}$ → $Noun_1$, "and", …$Noun_n$
- **Schema 8 for Rule 2:** «Concatenation ["W_{i+2} …W_n"; verb to be (plural); $data_{value(1)}$, "and", $data_{value(2)}$, "and", …$data_{value(n)}$] » :
  s → NP VP NP
  NP → "W_{i+2} …W_n"
  "W_{i+2} …W_n" → Det Noun PP
  PP → prep NP
  PP → AdjP prep Noun prep Number
  AdjP → Det Adjective Noun
  VP → verb to be (plural)
  NP → $data_{value(1)}$, "and", …$data_{value(n)}$
  $data_{value(1)}$, "and", …$data_{value(n)}$ → $Name_1$, "and", …$Name_n$
  $data_{value(1)}$, "and", …$data_{value(n)}$ → $Number_1$, "and", …$Number_n$
  $data_{value(1)}$, "and", …$data_{value(n)}$ → $Noun_1$, "and", …$Noun_n$

3.2.2 Surface planning

Once the content of our textual answer is determined and structured, it is necessary to transform the plan of this textual answer into a sequence of sentences. To accomplish this task, we try to convert the elements constituting the answers templates into lexical terms by referring to two functions which are lexicalisation and aggregation. Lexicalisation allows us to decide which specific words should be used to express the content (see Figure 17). For example, real names, verbs, adjectives and adverbs to appear in the textual answer are chosen from the WordNet lexicon.

Figure 17 A lexicalisation illustration

[sentence(np(lex = 'W_{i+1} … W_n'), vp(lex = be), np(lex = '$data_{value1}$', '$data_{value2}$'))]

⇩

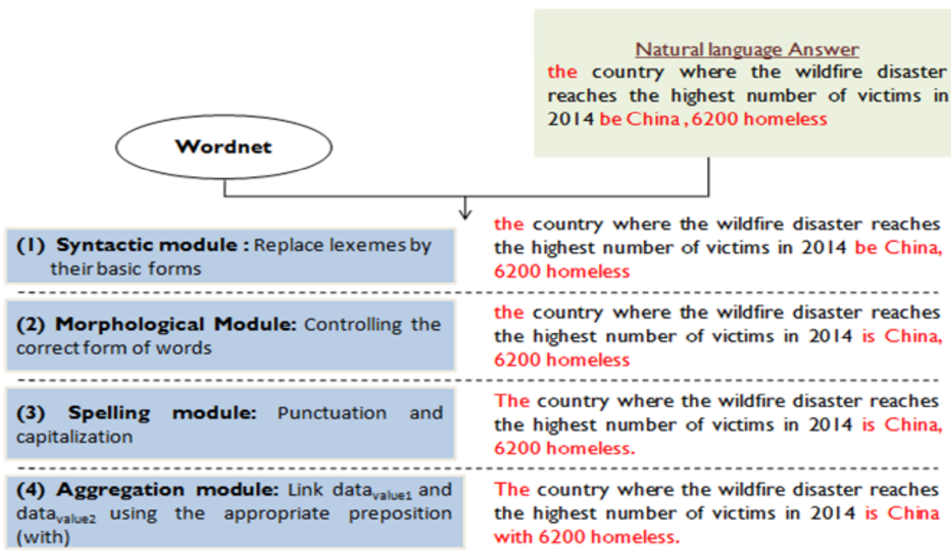
[sentence(np(lex = the country where the wildfire disaster reaches the highest number of victims in 2014), vp(lex = be, tense = present, number = singular), np(lex = China, 6200 homeless)]

The aggregation represents the way in which the structures created by the planning of our textual answer must be mapped onto linguistic structures such as sentences and paragraphs. For example, two ideas can be expressed as two separate sentences or as a single one [see Figure 18 – (4)].

3.2.3 Surface realisation

Once the elements of the answers templates are aggregated, lexicalised and grouped in a syntactic tree, the realisation can be carried out (see Figure 18). For this task, we present two essential functions. The first is the structural realisation; it consists in converting the sentence specifications by inserting annotations to reflect the structure of our textual answer. The second function is the linguistic realisation; it ensures the realisation of the syntactic structure in the textual answer (flexion, punctuation, layout...). This realisation is the basis of the use of grammatical rules. For example, for verbal syntagms we try to specify the following values: form: {bare infinitive, imperative, present participle}, tense: {past, present, future}, etc. For NPs we try to specify the following values: number: {singular, plural}, gender: {masculine, feminine}.

Figure 18 Structural and linguistic realisation (see online version for colours)



4 Simulation results

To implement our approach, we adopt the Java programming language. Using this language, we have developed STTDG (stands for spatio-temporal textual descriptions generation) interface as a stand-alone integrated into the geographic information system (GIS) OpenJUMP. For the sake of validation we are concerned with data gathered from the natural disasters database (NATDIS) (http://www.catnat.net). This latter, is a free GDB which could be replaced by any other database if it is available freely. In what follows we detail the different functionalities of the proposed STTDG tool.

4.1 The STTDG tool functionalities

The overall textual answer reformulation process is triggered once a user question is provided to the system querying the GDB for some information. In background the processing pipeline starts by pre-processing the question by exploiting the NLP techniques. To this end we make use of the Stanford CoreNLP (Manning et al., 2014) to recognise geographic entities and relations, to identify the POS tags and to extract dependency relations. More precisely, to recognise geographic entities and relations we adopted the generic framework TokensRegex (Chang and Manning, 2014) included in Stanford CoreNLP (Manning et al., 2014). This framework provides annotations based on regular expressions. Therefore, TokensRegex was used to develop SUTime (Chang and Manning, 2012) a rule-based temporal recogniser for detecting temporal expressions. The depicted items serve to generate ST-SQL statement and clauses throughout a set of pre-established heuristics (see Figure 19). Once the ST-SQL query is generated, it can be used to enquiry the GDB for the requested answers (see Figure 20). The latter will be fed in as the input of the answer reformulation stage. In order to reformulate the returned answers into a natural language answer (see Figure 21) the content of the answer to be generated has to be identified. Once the content is determined, it has to be structured and this by identifying the textual template to be translated into linguistic expressions. In this context, the templates will define different patterns of the textual answers and they will be instantiated according to a set of rules. Furthermore, a schema definition is necessary, in order to structure the content of the generated textual answer. Once the elements are ordered, it is appropriate to choose the words that will express them. The lexicalisation and aggregation are two functions used to convert the elements of information in order to generate answers in lexical terms using the WordNet lexicon. Finally, the realisation can be carried out. For this task we used the SimpleNLG developer, which is a Java API to create a syntactic text structure.

Figure 19 ST-SQL query generation from the ST_{uq} (see online version for colours)

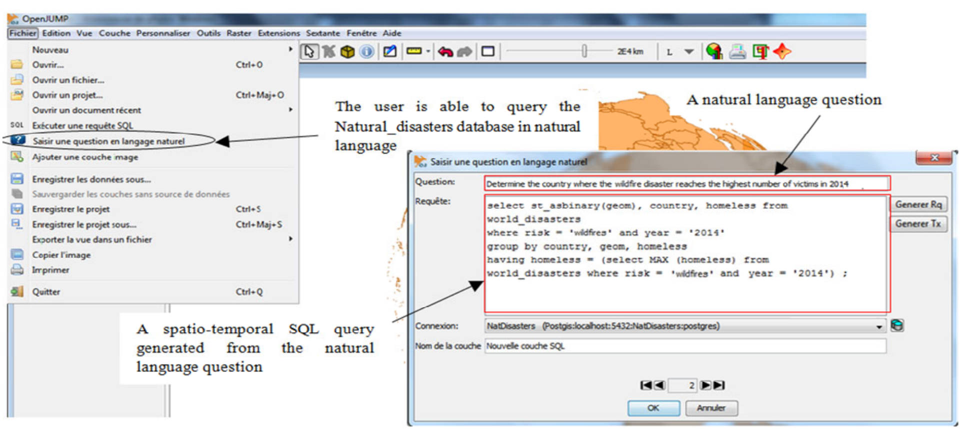


Figure 20 The extracted answers using the generated ST-SQL query (see online version for colours)

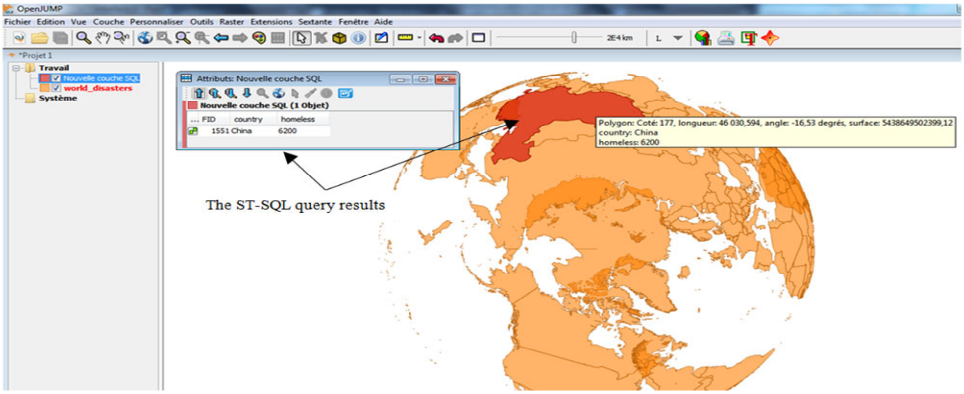
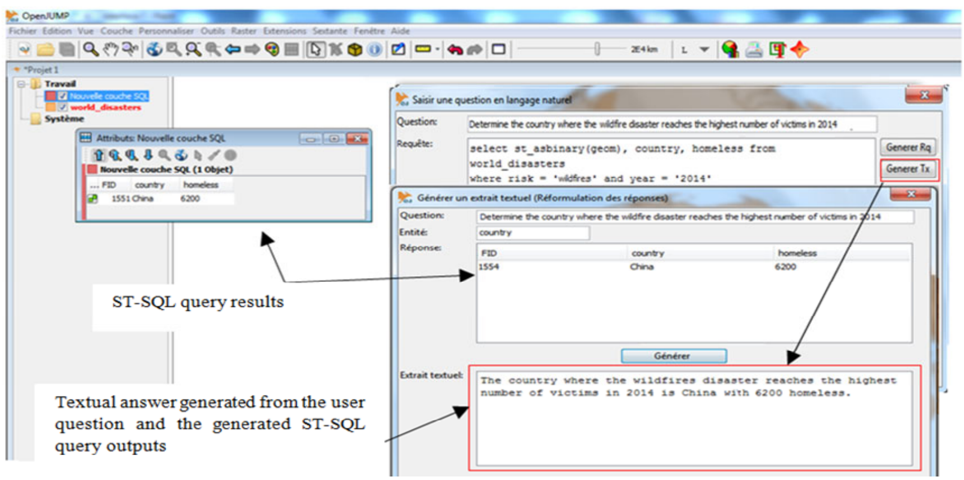


Figure 21 The answers reformulation stage (see online version for colours)



5 Evaluation

The evaluation is twofold. First we have to evaluate the performance of our system to generate a ST-SQL query and second we have to evaluate the performance of our system to generate a textual answer.

5.1 Evaluating system performance to generate a ST-SQL query

To evaluate the performance of our system to generate the right spatio-temporal query, we propose to use the two well-known measures: precision and recall. In fact, we evaluate if the spatio-temporal queries represent correctly the introduced questions. In this context, we provide a number of natural language questions to experts who master the geographic SQL query language to generate the appropriate queries. Then the system generated SQL queries from the provided natural language questions will be compared to those generated by the experts. Indeed, when a user introduces his question, our system can provide different types of results: a spatio-temporal query providing correct answer by referring to the expert generated query, a query which does not reflect the content of the natural language question because of a bad translation of the latter into a ST-SQL, and finally a third possible result is that the system is unable to ensure the translation process. In this context, the recall and precision measures can be defined as follows: the recall is the number of queries generated by our system and providing correct answers, relative to the number of generated spatio-temporal queries (ST-SQL). Precision represents the number of generated queries providing correct answers, with respect to the total number of the proposed questions.

$$\text{Recall} = \frac{\text{Number of generated ST-SQL queries providing correct answers}}{\text{Total number of generated ST-SQL queries}} \tag{2}$$

$$\text{Precision} = \frac{\text{Number of generated ST-SQL queries providing correct answers}}{\text{Number of proposed natural language questions}} \tag{3}$$

In the same context we adopted an additional metric used to evaluate the system effectiveness called willingness (Minock, 2010). This measure can be defined as the number of queries (ST-SQL) providing correct and incorrect answers, relative to the total number of the proposed natural language questions.

$$\text{Willingness} = \frac{\text{Number of ST-SQL queries providing correct and incorrect answers}}{\text{Number of proposed natural language questions}} \tag{4}$$

Based on these measurements, we prepared a set of 300 questions. The queries are checked manually. Different types of answers are returned when executing these queries. The results are given in Table 1.

Table 1 Experimental results

The total number of proposed questions	300
The number of spatio-temporal queries generated by our system	269
Number of questions that the system is unable to translate into spatio-temporal queries	31
Number of queries providing correct answers	232
Number of queries providing incorrect answers	37
Precision	86.2%
Recall	77.3%
Willingness	89%

Based on the recall (77.3%) and precision (86.2%) rates obtained in Table 1, we can notice that our system has managed to achieve satisfactory results as for the translation of users' questions into structured queries. we can infer that our system has achieved satisfactory results in translating user questions into structured queries. Thus, the obtained willingness rate is around 89%, which confirm the performance and the efficiency of our system.

5.2 Evaluating system performance to generate a textual answer

At this level, our goal is to evaluate the performance of our system to generate a textual answer. Our evaluation strategy consists in providing human experts all the necessary ingredients (the natural language questions and the extracted answers from the GDB) to propose understandable concise texts. Besides, the experts are solicited to use as much as possible the terms incarnated into the questions to minimise the disparity of their answers. Hence, the textual answer that makes the agreement of a maximum of experts is considered as a correct one. The latter will be considered as a reference to evaluate the system generated textual answer.

For the sake of evaluation we adopt the cosine, precision and recall measures used by the latent semantic analysis technique to analyse the lexical cohesion in texts. Thus, this technique is used in a growing number of NLP searches. This semantic space is used to estimate the semantic similarity between words, sentences, paragraphs and even texts. In this context, to evaluate the quality of our generated textual answer, we propose first, an *intra-text* evaluation. The latter consists in measuring the semantic similarities between the different adjacent sentences of the same textual answer. Then, we propose an *inter-text* evaluation allowing to compare our generated textual answer with the textual answer manuscript which made the agreement of the maximum of experts. In fact, the latent semantic analysis method was used to measure cosines between contiguous sentences. The average of these cosines is used as an estimation of the lexical cohesion of this textual answer. The cosine formula can be defined as follows:

$$\operatorname{Cos}\left(P_i, P_k\right) = \frac{\overrightarrow{P_j}\,\overrightarrow{P_k}}{\left|\overrightarrow{P_j}\right|\left|\overrightarrow{P_k}\right|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,k}}{\sqrt{\sum_{i=1}^{m} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,k}^2}} \tag{5}$$

This measure is used to calculate the semantic similarity between two sentences P_j and P_k where $w_{i,j}$ represents the weight of the term i in the sentence j, m and n are the numbers of respective terms of the two sentences. Besides, in order to measure the quality of the generated textual answer we make use of the: recall and precision metrics.

In the context of automatic text generation, the recall can be defined as the number of pairs of contiguous and semantically linked sentences, relative to the total number of sentences in the generated textual answer.

$$\text{Recall} = \frac{\text{Number of pairs of semantically similar contiguous sentences}}{\text{Total number of sentences in the textual answer}} \tag{6}$$

Regarding the precision, is defined as the number of semantically linked contiguous pairs of sentences found, relative to the number of pairs of contiguous sentences.

$$\text{Precision} = \frac{\text{Number of pairs of semantically similar contiguous sentences}}{\text{Number of pairs of contiguous sentences}} \tag{7}$$

A lower precision value indicates that few pairs of contiguous sentences are semantically similar. In this case, we do not risk having a semantic redundancy problem, and our system can be considered accurate. In the same context, a lower recall value indicates that the system produced a textual answer consisting of semantically rich sentences. Table 2 summarises the results found.

Table 2 Evaluation of the quality of our generated text answer

Indices	*Generated textual answer*	*Manuscript textual answer*
Number of sentences in a textual answer	10	11
Number of pairs of contiguous sentences	9	10
Number of pairs of semantically contiguous similar sentences	5	6
Average cosine between pairs of contiguous sentences (semantically similar)	0.18	0.30
Type/token ratio	0.49	0.45
Precision	55.55%	60%
Recall	50%	54.5%

In the experiment we have carried out, we find that the more diverse the vocabulary of a textual answer, the less the cosine between the pairs of contiguous sentences is high. This interpretation is justified by the correlation analysis between the lexical diversity measure, the type/token ratio (TTR) (the ratio between the number of different words and the total number of words in a text; the bigger it is and the more the vocabulary is rich) and the average cosine measure between contiguous sentences. First, we notice many differences between the two textual answers (generated textual answer and the handwritten text). Concerning the precision and the recall, we note that in our textual answer we detected nine pairs of contiguous sentences. Among these sentences, we confirm that five pairs are semantically similar. The relative precision is 55.55%, as well as the recall is 50%. Pairs of semantically non-similar sentences are those that do not share any word and whose cosine is equal to 0.

By comparing our textual answer with the other handwritten textual answers, we find that the more the vocabulary of a textual answer is semantically rich, the less the semantic similarity between adjacent sentences is high and the less the recall and the precision are high. Our experiments allowed us to judge the performance of our approach. This judgment is obtained by comparing our generated textual answer with a manuscript textual answer proposed by experts. According to this comparison, we noticed that a handwritten textual answer is semantically richer than an automatically generated one. However, this does not prevent our approach leading to acceptable results, while reflecting on improvements that enrich the semantics of our textual answers.

6 Conclusions

Along through this paper we have studied an approach motivated by the difficulties facing non-technical users while interacting with a GDB. The main idea of our proposal is to establish mappings from a natural language query to a SQL well formatted request to alleviate the GIS users from any SQL expertise requirements. Besides, a natural language translation of the retrieved answer is proposed as a final step. The overall process harnesses extensively NLP techniques. Hence, first the user question is interpreted to be translated into a set of statements and clauses as the main ST-SQL query components. This allows interacting with the GDB to retrieve the query response. To be understandable by the large communities of users, we propose to reformulate the extracted answer by generating the corresponding textual representation which helps users to interpret its meaning. The STTDG tool resulting from the implementation of our approach was integrated to the openJUMP GIS. Our approach achieves good results while conducting validation tests on disaster datasets.

Our approach can be used to help experts during their meetings and permit them to translate a spreadsheet of data (discussion, comments, which were unavailable, etc...) into a textual report which can be transmitted to everyone. In addition, websites which are concerned with geographic topics can be improved and optimised for search engines using our approach. These websites can help users searching for specific information and getting textual descriptions summarising the answers. In the same context some websites use maps to answer the user requirements. However, simple users does not have the potential to interpret those maps, in this case our approach can be a solution to improve these types of websites and provide other type of answer to describe the requested information in a more semantic way.

Given the satisfactory results of the current work, there is yet a significant room of improvement. Indeed, in the present paper we dealt only with two types of spatio-temporal queries: the simple spatio-temporal query and the spatio-temporal range one. Future works will address other types of spatio-temporal questions namely: spatio-temporal behaviour queries and spatio-temporal relationship queries. As far as the recognition of the geographic entities and relations is concerned, more advanced and efficient methods to recognise complex spatial and temporal relationships are to be conceived. Regarding the generation of the various clauses constituting a spatio-temporal query, all the steps can be improved by adding other clauses to improve the search of the information in the GDBs. Furthermore, we plan to improve the quality of the natural language reformulation of the retrieved responses by exploiting the rhetorical structure theory (RST).

References

Allen, J.F. (1984) 'A general model of action and time', *Artificial Intelligence*, Vol. 23, No. 2, pp.123–154.

Angeli, G., Liang, P. and Klein, D. (2010) 'A simple domain-independent probabilistic approach to generation', in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp.502–512.

Bahdanau, D., Cho, K. and Bengio, Y. (2015) 'Neural machine translation by jointly learning to align and translate', in *Proceedings of the International Conference on Learning Representations*. arXiv preprint arXiv:1409.0473.

Barzilay, R. and Lapata, M. (2005) 'Collective content selection for concept-to-text generation', in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pp.331–338.

Belz, A. (2008) 'Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models', *Natural Language Engineering*, Vol. 14, No. 4, pp.431–455.

Chang, A.X. and Manning, C.D. (2012) 'SUTIME: a library for recognizing and normalizing time expressions', in *8th International Conference on Language Resources and Evaluation*.

Chang, A.X. and Manning, C.D. (2014) *TokensRegex: Defining Cascaded Regular Expressions Over Tokens*, Stanford University Technical Report.

Danlos, L. (1989) 'Génération automatique de textes en langue naturelle', *Annales Des Télécommunications*, Vol. 44, pp.94–100.

Delpech, E. and Candillier, L. (2012) 'Nomao: un moteur de recherche géolocalisé spécialisé dans la recommandation de lieux et l'e-réputation', *TALN'2012*, Grenoble, France.

George, A.M. (1995) 'WordNet: a lexical database for English', *Communications of the ACM*, Vol. 38, No. 11, pp.39–41.

Goldberg, E., Driedger, N. and Kittredge, R. (1994) 'Using natural-language processing to produce weather forecasts', *IEEE Expert*, Vol. 9, No. 2, pp.45–53,

Green Jr., B.F., Wolf, A.K., Chomsky, C. and Laughery, K. (1961) 'Baseball: an automatic question-answerer', *Western Joint IRE-AIEE-ACM Computer Conference*, Vol. 19, pp.219–224.

Guessoum, D., Miraoui, M. and Tadj, C. (2016) 'A modification of Wu and Palmer semantic similarity measure', in the *Tenth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp.42–46.

Hunter, J., Freer, Y., Gatt, A., Reiter, E., Sripada, S. and Sykes, C. (2012) 'Automatic generation of natural language nursing shift summaries in neonatal intensive care: Bt-nurse', *Artif. Intell. Med.*, Vol. 56, No. 3, pp.157–172.

Kahn, M.G., Fagan, L.M. and Sheiner, L.B. (1991) 'Combining physiologic models and symbolic methods to interpret time-varying patient data', *Methods of Information in Medicine*, pp.167–178.

Lebret, R., Grangier, D. and Auli, M. (2016) 'Neural text generation from structured data with application to the biography domain', in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp.1203–1213.

Li, F. and Jagadish, H.V. (2014) 'Constructing an interactive natural language interface for relational databases', *VLDB*, pp.73–84.

Liang, P., Jordan, M.I. and Klein, D. (2009) 'Learning semantic correspondences with less supervision', in the *4th International Joint Conference on Natural Language Processing of the AFNLP*, pp.91–99.

Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J. and McClosky, D. (2014) 'The Stanford CoreNLP natural language processing toolkit', *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp.55–60.

Marneffe, M.C., MacCartney, B. and Manning, C.D. (2006) 'Generating typed dependency parses from phrase structure parses', *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*.

Mei, H., Bansal, M. and Walter, M.R. (2016) 'What to talk about and how? Selective generation using LSTMs with coarse-to-fine alignment', in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp.720–730.

Minock, M. (2010) 'C-phrase: a system for building robust natural language interfaces to databases', *Journal Data and Knowledge Engineering*, pp.290–302.

Pereira, F.C.N. and Warren, D.H.D. (1980) 'Definite clause grammars for language analysis – a survey of the formalism and a comparison with augmented transition networks', *Artificial Intelligence*, Vol. 13, No. 3, pp.231–278.

Raza, A. (2008) 'Deriving spatiotemporal relations from simple data structure', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. 37, pp.13–18.

Tan, J., Wan, X. and Xiao, J. (2017) 'Abstractive document summarization with a graph-based attentional neural model', in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp.1171–1181.

Van Deemter, K., Theune, M. and Krahmer, E. (2005) 'Real versus template-based natural language generation: a false opposition?', *Computational Linguistics*, Vol. 31, No. 1, pp.15–24.

Woods, W. (1977) 'Lunar rocks in natural English: explorations in natural language question answering', *Linguistic Structures Processing*, Vol. 5, pp.521–569, North Holland Publishing Company, Amsterdam.

Worboys, M.F. (1992) 'A generic model for planar geographical objects', *International Journal of Geographical Information Systems*, pp.353–372.

Wu, Z. and Palmer, M. (1994) 'Verb semantics and lexical selection', in *32nd Annual Meeting of the Association for Computational Linguistics*, pp.133–138.

Yaghmazadeh, N., Wang, Y., Dillig, I. and Dillig, T. (2017) 'Type and content-driven synthesis of SQL queries from natural language', *CoRR*.

Yuan, M. and McIntosh, J. (2002) 'A typology of spatiotemporal information queries', in Ladner, R., Shaw, K. and Abdelguerfi, M. (Eds.): *Mining Spatio-Temporal Information Systems*. *The Springer International Series in Engineering and Computer Science*, Vol. 699, Springer, Boston, MA.