

## Research Article

# Natural Language Processing and Fuzzy Tools for Business Processes in a Geolocation Context

Isis Truck<sup>1</sup> and Mohammed-Amine Abchir<sup>2</sup>

<sup>1</sup>CHArt Laboratory EA 4004, Paris 8 University, Saint-Denis, France

<sup>2</sup>Deveryware, Paris, France

Correspondence should be addressed to Isis Truck; [isis.truck@univ-paris8.fr](mailto:isis.truck@univ-paris8.fr)

Received 11 December 2016; Accepted 23 March 2017; Published 24 May 2017

Academic Editor: António Dourado Pereira Correia

Copyright © 2017 Isis Truck and Mohammed-Amine Abchir. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the geolocation field where high-level programs and low-level devices coexist, it is often difficult to find a friendly user interface to configure all the parameters. The challenge addressed in this paper is to propose intuitive and simple, thus natural language interfaces to interact with low-level devices. Such interfaces contain natural language processing (NLP) and fuzzy representations of words that facilitate the elicitation of business-level objectives in our context. A complete methodology is proposed, from the lexicon construction to a dialogue software agent including a fuzzy linguistic representation, based on synonymy.

## 1. Introduction

The question of geolocation is the core base of many companies working on geomarketing, traffic planning, or transport logistics. In the geomarketing area, social networks with recommendations (nearby social events, nearby restaurants, etc.) and determination of a customer profile depending on his/her choices, preferences, tastes, and so forth are examples of growth sectors, especially with the expansion of the mobile market that is rapidly growing. The company we are working with is a leader in geolocation middleware and proposes systems for child location, vehicle and fleet tracking, or delivery rounds. From the company's point of view, persons, vehicles, and goods are considered as devices that have to be tracked with accuracy using a *Geohub* and notifications must be sent according to the type of the required tracking. On the contrary, from the customer's point of view, a service has to be proposed according to a prior agreement but no (or few) technical detail(s) has (have) to be known. The *Geohub* coordinates geoinformation on a single platform to track devices and to give them orders knowing their position (e.g., what to do in case of traffic jam or accident on the road or if the vehicle has broken down). The *Geohub* interacts with all the tracked devices. Low-level messages (written in the Forth

language) are sent by both the devices and the *Geohub* in "push" or "pull" mode. Two roles have to be distinguished: (1) the company that sells and connects devices to its hub and that configures the hub to satisfy the customer and (2) the customer who expresses his/her need (e.g., his/her fleet tracking) to an employee of the company that is able to configure the hub. Thus, the employee has to understand the customer's needs and elicit his/her preferences while configuring the *Geohub*.

An important challenge is to propose a smart interface, smart enough to let the customers configure the *Geohub* by themselves. The customers would have a phone conversation with a virtual assistant. We know that natural language processing (NLP) deals with NP-complete problems, which is why such an interaction needs many contextual elements to limit the possibilities. However, even with context, the problem remains quite hard since it implies important expertise to be able to translate the needs expressed in a natural language into a set of Forth scripts and programs written in other languages. Thus, there is a need for tools to capture these sometimes imprecise requirements (e.g., "I would like to be notified when one of my fleet vehicles arrives near the warehouse") and transform them into business processes. The fuzzy logic and its various methods and tools are of great

help for such needs and especially the methods that deal with linguistic variables.

In previous works, we proposed a linguistic model designed to better understand and meet the needs expressed in a natural language: the *semantic fuzzy 2-tuple model*. While a previous paper [1] proposed a classification of several linguistic models, including the semantic fuzzy 2-tuple model, this paper focuses on intuitive and simple interfaces, expressed in natural language, in a real-case study (geolocation applications), using the semantic fuzzy 2-tuple model.

The paper is organized as follows: we first review some NLP techniques and we focus on an interesting linguistic fuzzy model to deal with imprecisions. Our approach that mixes NLP with fuzzy tools in the geolocation context is then explained. Finally, a use case is presented: it exhibits the use of a fuzzy semantic approach to activate an alert in a geolocation context.

## 2. State of the Art

We first review some NLP techniques and then we are interested in fuzzy logic that deals with *computing with words*.

*2.1. Some NLP Techniques.* Since the first techniques to deal with natural language, many methods and algorithms have been proposed to understand and disambiguate sentences [3–6]. For example, decision trees and statistical models may give good results as soon as there is context enough in the speech corpus. In NLP, a formal grammar draws a set of normative rules to describe how the natural language runs.

Among the various formal grammars are the generative grammars, the transformational grammars, and the tree-adjoining grammar (TAG) [7]. To make discourse analysis, part-of-speech tagging is often used to disambiguate words (e.g., “display” can be a noun, a transitive verb, or an intransitive verb) [8]. Syntax is part of formal grammar that deals with rules for the structure of a string (distribution of words, noun and adjective agreements, etc.). Meaning and formal syntax are the basis of the comprehension, according to Chomsky [9].

Moreover, there have been a lot of works about semantic parsing and question answering. For example, Popescu and collaborators developed semantic parsing for querying databases [10], Berant et al. trained a semantic parser on a web database, learning from question-answer pairs [11], and Shi and Mihalcea integrated several lexical resources into a unified knowledge base to enable robust semantic parsing [12].

Our aim is a bit different: we aim at building a (rather small) business-oriented database, depending on the needs of customers and on the way the experts define the business processes. However, semantics is the focus of our concerns because we feel with natural language. It is well known that semantics is the study of the meaning of words or sentences in their context. The concept of meaning is often fuzzy because natural language may be imprecise if we consider that each word (or set of words) has more than one meaning (without taking into account the figurative sense that is something else again).

Moreover, synonymy can give clues to understand the meaning of an expression or to link two expressions. Synonymy is one of the most difficult semantic relations to handle in NLP. It is very complex because no standard metric is available to measure the distance or proximity between two words or two concepts. In [13], some authors propose to quantify synonymy by what they call the *angular distance* and they recognize two types: *relative* synonymy and *subjective* synonymy. This distance is defined as a semantic distance between two conceptual vectors, where a conceptual vector is a combination of several concepts. Thus, relative synonymy is the synonymy between two terms with respect to a given concept. This synonymy is transitive with respect to the concept. Subjective synonymy permits stating that two terms are synonyms (while relative synonymy would have to state that they are not) when adopting a more general semantic field. For example, *to cut* and *to fragment* would not be relative synonyms but they would be subjective synonyms, from the general field “factory.”

In linguistics, the meaning of a vague or imprecise expression is a fuzzy set in the proper sense. What we call *fuzzy semantics* is an approach that uses fuzzy logic (Zadeh’s sense) to express the fuzziness of the meaning.

*2.2. Computing with Words and Fuzzy Logic.* Zadeh was the pioneer researcher to propose approaches to deal with imprecision and uncertainty when he introduced the fuzzy set theory, the fuzzy logic, and the concept of linguistic variables [14]. Since then, many fuzzy models have been proposed for *computing with words* but one seems the most appropriate in our case because it makes a simple correspondence between words and fuzzy scales: the 2-tuple fuzzy linguistic model [15].

Much work has been completed using the fuzzy linguistic approach and the 2-tuple model in many situations: distributed agents [16], genetic learning [17], industrial engineering [18], multicriteria decision-making [19], fuzzy decision tools [20], human resource management [21], and so forth.

In this model, each word or linguistic expression is translated into a linguistic pair  $(s_i, \alpha)$  where  $s_i$  is a triangular-shaped fuzzy set and  $\alpha$  is a symbolic translation. If  $\alpha$  is positive, then  $s_i$  is reinforced; else,  $s_i$  is weakened. If the information is perfectly balanced (i.e., the distance between each consecutive word is exactly the same), then all the  $s_i$  values are equally distributed on the axis. But if not (which may happen when talking about distance; e.g., “almost arrived” and “close to” are closer to each other than “far” and “out of route”), the  $s_i$  values may not be equally distributed on the axis.

Another model has thus been proposed to deal with such information called multigranular linguistic information [2]. To perform the computations, linguistic hierarchies composed of an odd number of triangular fuzzy sets of the same shape, equally distributed on the axis, are used as a fuzzy partition. The way the levels of hierarchy are constructed is as follows: let  $t$  be the level and  $n(t)$  be the number of triangular fuzzy sets in the partition. Given the initial term  $n(1) = 3$ , each subsequent term  $n(t + 1)$  is determined by the relation  $n(t + 1) = 2 \times n(t) - 1$ . Thus,  $n(2) = 5$ ,  $n(3) = 9$ ,  $n(4) = 17$ , and so forth (see Figure 1).

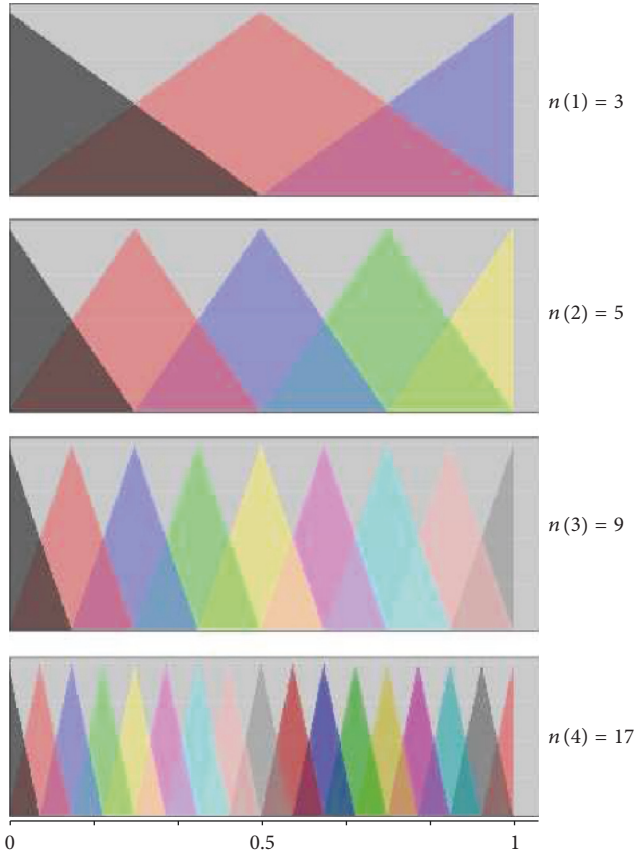


FIGURE 1: Linguistic hierarchies with 3 consecutive levels, according to [2].

A word may have one or two linguistic hierarchy(ies) and the representation of a notion (such as “distance”) can be made up of several hierarchies. Thus, the fuzzy sets obtained are still triangular-shaped but not always isosceles triangular-shaped and the notation  $s_i^j$  permits keeping both the hierarchy and the linguistic term. See [22, 23] for a deeper review of these models.

Nevertheless, we have shown in recent papers that the 2-tuple model with unbalanced linguistic term sets does not allow modeling any kind of unbalanced scale. Indeed, when one linguistic expression is very far away from its next neighbor, the 2-tuple fuzzy linguistic model puts them artificially closer to each other because no appropriate linguistic hierarchy level can be found [24, 25].

Recently, the same team has proposed an improvement of the 2-tuple model with *extended linguistic hierarchies* (ELH) [26]. The ELH permit generating the “missing” levels of hierarchies (i.e., level  $t_1$  with  $n(t_1) = 7$ , level  $t_2$  with  $n(t_2) = 11$ , level  $t_3$  with  $n(t_3) = 13$ , etc.). However, this appears to be too constraining because we loose the flexibility of the original model. Indeed, when adding so many labels on the axis, how can we guarantee that each label will correspond to a linguistic term or expression? The idea behind the fuzzy 2-tuple model is to use the continuity of the scale while computing with only a few words (seen as discrete numbers

on the axis). With the ELH, this is hard to do. However, we have recently proposed a comparison between two 2-tuple linguistic models and we showed the links between them [27].

The model we have proposed in a previous paper is called *semantic fuzzy 2-tuple model* and is of course inspired by the 2-tuple fuzzy linguistic model [28]. It uses the symbolic translations  $\alpha$  to generate the data set. In our model, the 2-tuples are twofold. Except the first one and the last one of the partition, they all are composed of two half 2-tuples: an upside and a downside 2-tuple. In our context (geolocation) where the linguistic terms are usually unbalanced, this choice is quite relevant and it has been shown in a previous paper [29].

**2.3. Semantic Fuzzy 2-Tuples.** More precisely, the linguistic values are composed by a pair  $(s, v)$  where  $s$  is a linguistic term and  $v$  is a number giving the position of  $s$  on the axis.

*Definition 1* (see [28]). Let  $\mathcal{S}$  be an unbalanced ordered linguistic term set and  $U$  be the numerical universe where the terms are projected. Each linguistic value is defined by a unique pair  $(s, v) \in \mathcal{S} \times U$ . The numerical distance between  $s_i$  and  $s_{i+1}$  is denoted by  $d_i$  with  $d_i = v_{i+1} - v_i$ .

*Definition 2* (see [28]). Let  $S = \{s_0, \dots, s_p\}$  be an unbalanced linguistic label set and  $(s_i, \alpha)$  be a linguistic 2-tuple. To support the unbalance,  $S$  is extended to several balanced linguistic label sets, each one denoted by  $S^{n(t)} = \{s_0^{n(t)}, \dots, s_{n(t)-1}^{n(t)}\}$  defined in the level  $t$  of a linguistic hierarchy LH with  $n(t)$  labels. There is a unique way to go from  $\mathcal{S}$  (Definition 1) to  $S$  according to an algorithm detailed in [28].

*Definition 3* (see [28]). Let  $l(t, n(t))$  be a level from a linguistic hierarchy. The *grain*  $g$  of  $l(t, n(t))$  is defined as the distance between two 2-tuples  $(s_i^{n(t)}, \alpha)$ .

The grain  $g$  of a level  $l(t, n(t))$  is obtained as  $g_{l(t, n(t))} = 1/(n(t) - 1)$ . For instance, the grain of the second level is  $g_{l(2,5)} = 0.25$ . The grain  $g$  of a level  $l(t - 1, n(t - 1))$  is twice the grain of the level  $l(t, n(t))$ :  $g_{l(t-1, n(t-1))} = 2g_{l(t, n(t))}$ .

The aim of the partitioning is to assign a label  $s_i^{n(t)}$  (indeed one or two) to each term  $s_k$ . The selection of  $s_i^{n(t)}$  depends on both the distance  $d_k$  and the numerical value  $v_k$ . We look for the nearest level—they are all known in advance—that is, for the level with the closest grain from  $d_k$ . Then, the right  $s_i^{n(t)}$  is chosen to match  $v_k$  with the best accuracy.  $i$  has to minimize the quantity  $\min_i |\Delta^{-1}(s_i^{n(t)}, 0) - v_k|$ , with  $\Delta^{-1}(s_i, \alpha)$  being the function that computes the numerical equivalent value of  $(s_i, \alpha)$ ; that is,  $\Delta^{-1}(s_i, \alpha) = i + \alpha$ .

By default, the linguistic hierarchies are distributed on  $[0, 1]$ , so a scaling is needed in order for them to match the universe  $U$ .

The function returns a set of bridge unbalanced linguistic 2-tuples with a level of granularity that may not be the same for the upside rather than for the downside.

We now explain how NLP tools have been used to enhance business processes in a geolocation context.

### 3. NLP and Fuzzy Linguistics

Offering an interface based on a full natural language dialogue is a complex task that is NP-hard. Thus, reducing the complexity is compulsory. The semantic analysis implies using a substantial database to catch the meaning of the dialogue and to take into account the imprecision and subjectivity due to the natural language and human reasoning.

In the following, we propose an approach to elicit the choices (in a natural language interface), so the experts can express their business process needs: configure the GPS mobiles and create geolocation alerts easily.

*3.1. Natural Language Integration.* We aim at giving customers and experts a way to express their needs and business objectives *via* a natural language interface. In the short term, the idea is to obtain a complete vocal assistant in the geolocation context. That assistant would permit retrieving information about the condition of a user's mobile or, more generally, of the mobiles of a user's fleet. It would also permit changing the mobile configuration, programming alerts, and offering high-level features that would be translated into low-level configurations and/or alert programming.

The approach we propose follows the *Question Answering Systems* in which the program (the software) leads the dialogue to avoid dead ends or unproductive loops. The program starts with a first question. After each answer, it analyzes it to decide whether it shall make a decision right now or ask another question, according to a given scenario.

The first step of the dialogue interface construction is to prepare a lexicon to be able to analyze the sentences.

The lexicon is built from a corpus of documents among which are advertisements from the company, some documentation of the web applications and of the *Application Programming Interface* (API), and language files (if the interface is conceived and designed for several countries).

In software engineering, texts displayed in the applications as well as the labels, the buttons, the menu titles, the help menu, and so forth are grouped together in a single "key-value" file for each supported language. Values are dynamically loaded while the software is being started and the language file chosen is the one that the user speaks. This permits separating style (text) and content (code) and facilitating the software translations.

Once the corpus is exhaustive, a statistical study is performed on the occurrences of the words from the corpus. Of course, grammatical words are excluded. We obtain a set of the most relevant words of the jargon of the field (here: geolocation jargon); these words will constitute the entry-level lexicon. The method TF-IDF (Term Frequency-Inverse Document Frequency) [30] is completely efficient for this task.

The words obtained are now called *tokens*.

*3.2. Parts of Speech Tagging, Lexicon, and Synonyms.* Tokens, expressions, and parts of speech are grouped together in an XML file through PoS (parts of speech) tagging [31] where each lexicon input receives a set of descriptive tags.

Each input is labeled with a grammatical tag to express the grammatical category (noun, verb, adjective, adverb, etc.). These tags may help to disambiguate certain words (e.g., "base" can be an adjective, a noun, or a transitive verb) since this kind of ambiguity is very common in a natural language [8].

PoS tagging is realized through automatic tagging tools such as TreeTagger (<http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/>). TreeTagger permits regrouping the PoS tagging and the lemmatization: it groups together the different inflected forms of a word so they can be analyzed as a single item. Then, we add semantic tags to the most relevant lexicon inputs, from a business-level point of view. These tags permit representing the knowledge and the semantic concepts. Thus, several inputs (words) can be linked to these semantic tags. The empirical way to build the lexicon is unavoidable and necessary because expert knowledge is needed.

The following example shows a subset from the lexicon. Let us take the following need: "I want to receive an alert when the truck gets very close to the warehouse". The tagging permits obtaining tokens with a certain grammar (*gram*) and meaning (*sem*). "I" is grammatically a pronoun and has no particular meaning for us (it simply designates the customer). "alert" is grammatically a noun and has meaning ALERT. "gets" is grammatically a verb and has meaning ZONE\_ENTRY. "very" is grammatically an adverb and has meaning FUZZY\_MODIF\_+. "close to" is grammatically an adjective and has meaning DISTANCE.

```
<?xml version="1.0" encoding="UTF-8"?>
  <tokens>
    <token gram="PRON">I</token>
    <token gram="VERB">want</token>
    <token gram="VERB">to receive</token>
    <token gram="DET">an</token>
    <token gram="NOUN" sem="ALERT">alert</token>
    <token gram="CONJ" sem="ALERT_COND">when</token>
    <token gram="DET">the</token>
    <token gram="NOUN" sem="VEHICLE">truck</token>
```



```

<token gram="VERB" sem="ZONE_ENTRY">gets</token>
<token gram="ADV" sem="MODIF_FUZZY">very</token>
<token gram="ADJ" sem="DISTANCE">close</token>
<token gram="ADP">to</token>
<token gram="DET">the</token>
<token gram="NOUN" sem="POI">warehouse</token>
</tokens>

```

Semantic tags may represent generic elements (vehicles, persons, cities, etc.) as well as specific geolocation elements (types of alerts, points of interest, kinds of mobiles, etc.).

Now that the tagged basic core of the lexicon is created, it has to be enriched for a better exhaustivity and in order to make the interface accept the largest number of terms. As we are in a closed domain, the lexicon size will not be too big. The extension of the core is performed using a method based on several French dictionaries of synonyms:

- (i) DES (<http://www.crisco.unicaen.fr/des/>) (for *Dictionnaire Électronique des Synonymes*, i.e., electronic dictionary of synonyms)
- (ii) Dictionary of synonyms from Reverso (<http://dictionnaire.reverso.net/francais-synonymes/>)
- (iii) Dictionary of synonyms (<http://www.dictionnaire-synonymes.com/>)
- (iv) Dictionary of synonyms of Blue Painter (<http://www.synonymes.com/>)
- (v) Dictionary of the *Institut des Sciences Cognitives* (ISC) (<http://dico.isc.cnrs.fr/dico/fr/chercher>)
- (vi) Dictionary of synonyms of SenseGates (<http://www.sensagent.com/>)
- (vii) Dictionary of synonyms of the *Centre National de Ressources Textuelles et Lexicales* (CNRTL) (<http://www.cnrtl.fr/synonymie/>)
- (viii) Dictionary of synonyms of Synonymo (<http://synonymo.fr/>)
- (ix) Larousse dictionary (<http://www.larousse.fr/dictionnaires/francais>)

DES has one big advantage: it presents the synonyms as *cliques*. A *clique* is a subset of synonyms sharing a particular meaning [32, 33]. Thus, a word belongs to just as many cliques as it has different meanings.

For each term of the lexicon, a set of synonyms from the different sources is retrieved. The set is the *intersection* of all the synonyms to avoid the noise (i.e., too specific or too wide synonyms or synonyms with a figurative sense) and to avoid conflicts between dictionaries (e.g., term  $A'$  is a synonym of  $A$  according to dictionary 1, but  $A'$  is a synonym of term  $B$  according to dictionary 2: in that case,  $A'$  is excluded). This set is assigned with the semantic tags of the original term of the lexicon.

Of course, multilingual extensions could be added. The method would be the same, except for the dictionaries. For instance, if we consider German, we can use dictionaries such as <http://synonyme.woxikon.de/synonymliste>. If we use English, there are many dictionaries of synonyms, such as the following:

- (i) Thesaurus (<http://www.thesaurus.com/>)
- (ii) Dictionary of synonyms of Blue Painter (<http://www.synonymy.com/>)
- (iii) Dictionary of Oxford (<https://en.oxforddictionaries.com/thesaurus/>)
- (iv) Dictionary of Collins (<https://www.collinsdictionary.com/dictionary/english-thesaurus/distance>)

A bit like DES, the Dictionary of Collins has an advantage: main synonyms can be distinguished from additional synonyms, so there is a classification between synonyms, according to their semantic distance to the original word.

*3.3. Fuzzy Partitions to Catch the Meaning of the Concepts.* For each concept whose meaning may be imprecise or fuzzy, a fuzzy partition with our 2-tuple model is performed by experts. As an example, if the expert gives five labels to represent the semantic concept DISTANCE, they are by default considered as uniformly distributed on their axis. Looking for synonyms gives a set of five synonym bags, one bag per label. The distance between each label is computed using the number of shared synonyms in each bag. Resemblance rate matrices are then computed to determine both the *order* of the terms and the *distance* between them. The more common the synonyms, the closer. Finally, it is easy to construct the partition of the five unbalanced terms thanks to our 2-tuple linguistic model. See Figure 2 where both partitions (top: Herrera and Martínez's one; down: ours) are shown.

At the top of the figure, the partition seems perfectly balanced because the model forces having a midterm (here, "CloseTo"). The rest of the terms are automatically placed from left to right. At the bottom of the figure, our partition is unbalanced because we used our method with semantics and synonyms. Of course the big difference between both models is that in our case fuzzy subset "Far" meets "OutOfRoute" with a membership degree less than 0.5 (0.15 actually). But our model guarantees the minimal coverage property, as proved in [28], which is enough in the computations.

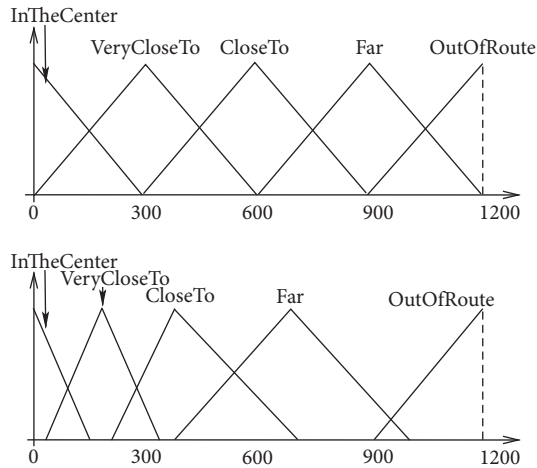


FIGURE 2: Unbalanced linguistic term sets: example of our partition model for distance.

The semantic tokens (e.g., “close to”) are then expressed through our 2-tuples and compared to the fuzzy partition.

```

FUNCTION_BLOCK Alert1
VAR_INPUT Battery:LING; Distance:LING; TimeTolerance:LING;
END_VAR
VAR_OUTPUT
    AlertTrigger:REAL;
END_VAR
FUZZIFY Battery
    TERM S:= pairs (Minimum, 0.0) (VeryLow, 10.0)
                (Low, 20.0) (Medium, 50.0)
                (High, 60.0) (VeryHigh, 80.0)
                (Maximum, 100.0);
END_FUZZIFY
FUZZIFY Distance
    TERM S:= pairs (InTheCenter, 0.0)
                (VeryCloseTo, 200.0)
                (CloseTo, 400.0) (Far, 700.0)
                (OutOfRoute, 1200.0);
END_FUZZIFY
FUZZIFY TimeTolerance
    TERM S:= pairs (Minimum, 0.0) (Medium, 60.0)
                (Maximum, 120.0);
END_FUZZIFY
DEFUZZIFY AlertTrigger
    TERM NoAlert:= trian 0.0 0.0 1.0;
    TERM Alert:= trian 0.0 1.0 1.0;
    METHOD: COG; // 'Center Of Gravity'

```

Adverbs such as “very” will modify their associated 2-tuple through the  $\alpha$  translation value, which has to be seen as a *modifier*.

In the following, we propose a use case using both fuzzy models: Herrera and Martínez’s one and ours.

As a use case, we keep the same example as before with an alert that is triggered depending on (1) the distance between the mobile and the arrival (a warehouse) that is subject to (2) time tolerance and depending on (3) the battery level. If the time tolerance is high enough, the alert is not triggered when the mobile is quite far away from the arrival and the battery level is low (to save battery). The alert is triggered, even if the battery level is low, when the mobile drives out of route of the arrival. The software and the library to compute the partitions have been written in Java, using jFuzzyLogic with the extension we have proposed (see the website of our project <https://salty.unice.fr/attachments/175/jFuzzyLogicExtended.jar>) and the FCL (Fuzzy Control Language) specification (IEC 61131-7). The FCL script below uses three inputs and one output.

```

END_DEFUZZIFY

RULEBLOCK Rules
    RULE 1: IF Battery IS Minimum AND Distance IS
           InTheCenter AND TimeTolerance IS
           Minimum
           THEN AlertTrigger IS NoAlert;
    ...
    RULE 28: IF Battery IS Maximum AND Distance IS Far
            AND TimeTolerance IS Minimum
            THEN AlertTrigger IS Alert;
    ...
    RULE 63: IF Battery IS Maximum AND Distance IS Far
            AND TimeTolerance IS Medium
            THEN AlertTrigger IS Alert;
    ...
END_RULEBLOCK

END_FUNCTION_BLOCK

```

Two different partitions were used for the inputs: the 2-tuple partition by Herrera et al. and our 2-tuple partition (in the FCL extract, only our partition is shown through the pairs). A study and a comparison between both models show that no alert is triggered with Herrera et al.'s model when (i) distance is between *CloseTo* and *Far* (=600) and battery level is *Maximum* (=100) and when (ii) distance is *Far* (=700) and battery level is *Maximum* while an alert would have been triggered with our 2-tuple model. This is due to the fact that the unbalancement is weaker in this model than in ours.

Figure 3 describes the exchanges between mobiles and the Geohub.

**3.4. Business Grammars.** Regarding the natural language interface, the next step is to determine a noncontextual business grammar. This grammar defines the business concepts taken into account by the dialogue interface. For example, in a geolocation alert, the grammar defines the various components of the alert and these components will be transformed in parameters or questions to ask users during the dialogue. This permits specifying a general concept by subconcepts that can themselves be specified later. The syntax used is of EBNF (Extended Backus-Naur Form) kind to define each element of the business grammar. The following example shows a simplified grammar for the definition of a geolocation alert:

```

ALERT = TYPE, MOBILE, PLACE, NOTIFICATION
MOBILE = VEHICLE | PERSON
TYPE = ZONE_ENTRY | ZONE_EXIT | CORRIDOR

```

```

PLACE = TOWN | ADDRESS | POI | ZOI
NOTIFICATION = RECIPIENT, MESSAGE
RECIPIENT = TEL_NUMBER | EMAIL
MESSAGE = MSG_TEXT

```

This grammar defines an alert as being composed (compulsory) of a type of alert, a mobile, a place, and a notification. A mobile is either a vehicle or a person, and so forth. In the example, POI means “point of interest” and ZOI “zone of interest.”

The grammar will be useful for leading the dialogue scenario (see Section 4.2).

Terminal symbols of each grammar are linked directly to one (or more) semantic tag of the business lexicon. Thus, the dialogue interface links the words from the lexicon and their semantics from the business level. For example, in the extract of grammar, a MOBILE is either a VEHICLE or a PERSON. Using the lexicon, we will find the entry *truck* tagged with semantic tag VEHICLE. The dialogue interface will make the connection with VEHICLE each time the word *truck* is found during the analysis of the user sentences.

## 4. Towards an Approach of a Dialogue Software Agent

The dialogue interface in natural language we propose is a kind of software agent.

**4.1. The Software Agent.** The goal of the agent is to interact with the user to elicit his/her choices and preferences and

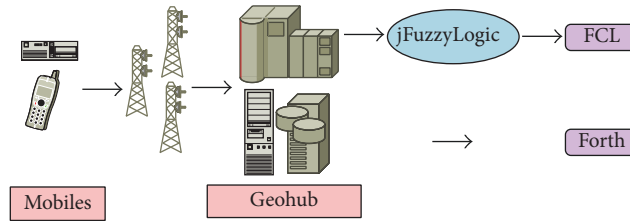


FIGURE 3: From mobiles to the Geohub: exchanges and use of FCL and Forth language.

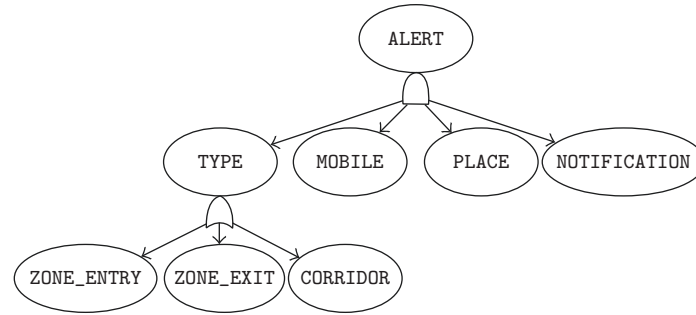


FIGURE 4: TAG representation (extract) for a business grammar.

then to realize the tasks that follow, such as creating an alert on the Geohub, configuring a mobile device, and displaying information of a user account. The agent uses the business lexicon and grammar that it loads dynamically while launching. The lexicon is stored in a hash table while the grammar is loaded through a TAG (tree-adjoining grammar). Despite the fact that the tree representation is restrictive regarding the knowledge representation, the tree is easier to use for immediate processing through database querying. In case of ungrammatical statements, the sentence is analyzed without any error because only the semantic tags are retrieved. If no semantic tag is detected, the agent asks a question to obtain those tags. The generation of answers is a wide subject [34] and is in our case performed from the template. Templates are composed of three kinds of elements: elements coming from the previous user statements, predefined expressions, and the answer. Thus, if the previous statement is “I want to notify Alan,” it is analyzed as “I want <notification> to notify</notification> <recipient>Alan</recipient>”. The agent will answer with a question generated by the following template: “What message do you want to send <recipient>?”, that is, “What message do you want to send Alan?” or “What message do you want to send your recipient?”

Figure 4 shows an extract of the grammar represented by a TAG.

The dialogue between the software agent and the user is composed of several steps:

- (i) The agent starts a conversation with an open question such as “Hello, what do you want to do?”
- (ii) It retrieves the answer and forwards it to the parser. The parser acts like a preprocessor that prepares the analyze phase. It labels each word of the sentence (the answer) thanks to the semantic tags of the lexicon.

- (iii) The next step is the analysis of the sentence. Only semantic tags are used (but the grammatical ones may be useful in future works) because there is a need for reducing the complexity (we recall that the aim is to catch the meaning) and because it permits some flexibility (ungrammatical sentences may be considered as correct as soon as the meaning is clear; see the principle of extended grammar by Harris [35]).
- (iv) Given that the nodes of the TAG correspond to components (or necessary conditions) of the accomplishment of a given task, the agent tries to check/validate all the nodes. The children nodes are linked together with an AND operator; that is, they all are compulsory to validate the parent node. The leaves of the same node are linked together with an OR operator; that is, only one leaf is needed. Each time a parameter is entered, the corresponding node is marked (validated).
- (v) Then, the agent checks whether all the nodes are marked:

- (a) If this is the case, then the whole set of parameters has been retrieved and the task may begin.
- (b) Else, the agent asks a question that corresponds to the missing parameter. Then, it parses and analyzes the answer and restarts the validation process.

The dialogue proceeds until all the parameters are entered by the user.

*4.2. Prototype Implementation.* We have developed a prototype software agent for a dialogue in natural language



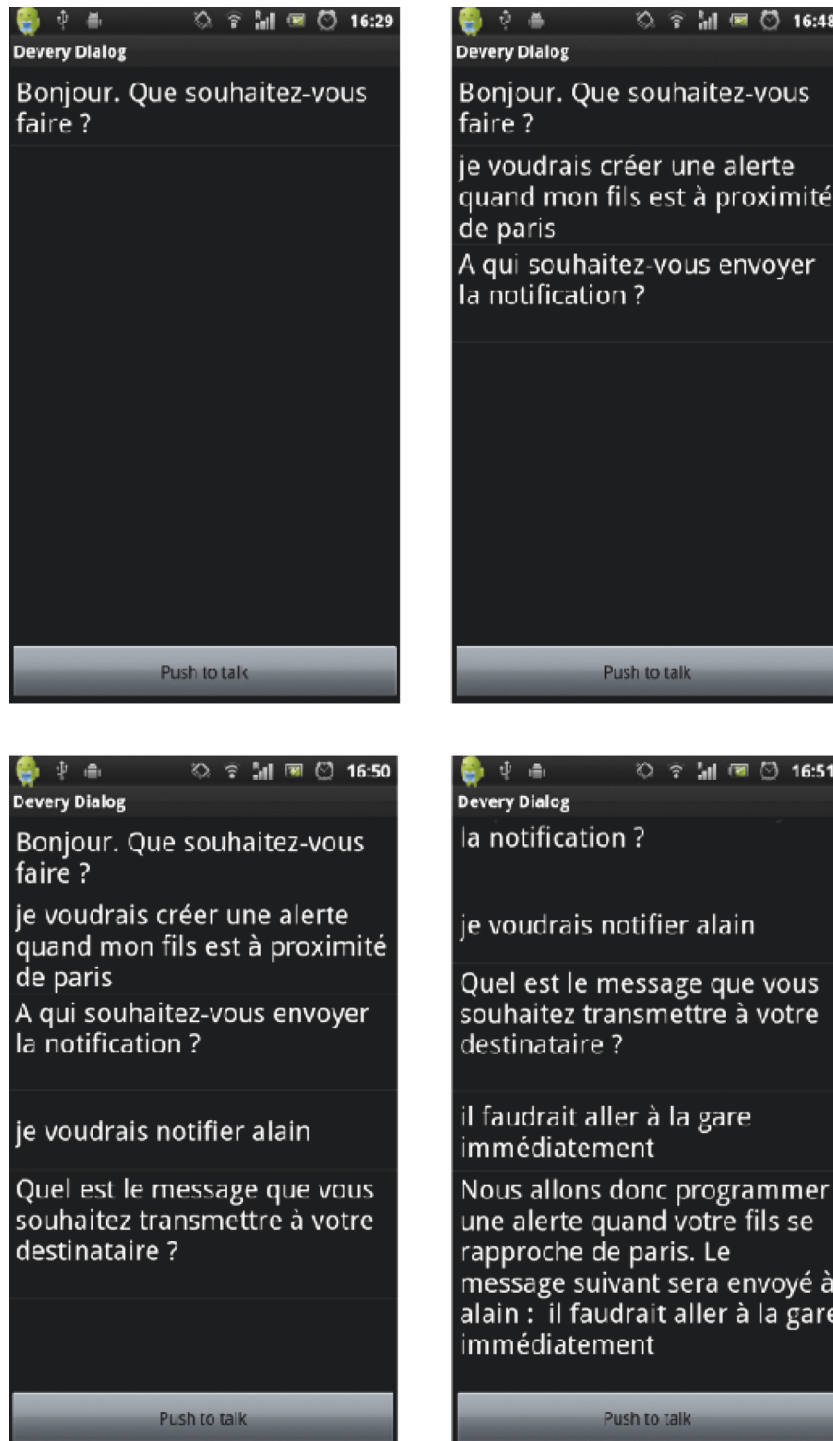


FIGURE 5: Example of a dialogue on Android platform (in French, the translation being included in the article).

specialized for geolocation in an Android application. To facilitate the dialogue, we used the speech recognition and synthesis of Android system for the user interaction. The application called DeveryDialog permits realizing several tasks like programming an alert, displaying the mobile list for a count, changing the frequency of data (GPS positions) collection, and so forth.

Figure 5 shows a short dialogue (in French) in four screens (from left to right and from top to bottom). Screen 1 is the welcome message (“Hello, what do you want to do?”). Screen 2 shows the wish of the user (“I would like to create an alert when my son gets close to Paris”) and the answer of the software agent (“Who do you want to send the notification to?”). The wish (sentence) of the user is annotated by the



FIGURE 6: Real-case example where an alert is triggered shortly after a mobile leaves the corridor.

agent, and then the business grammars are checked. In this case (alert creation), the *alert* TAG will be chosen. The agent tries to validate the tree elements since the user has given several parameters in a single answer. Indeed, he/she gave information about the alert type (*gets* is linked by semantic tag with *ZONE\_ENTRY* type), the mobile, and the destination. Thus, the agent makes a try about the notification in order to retrieve all the parameters (“Who do you want to send the notification to?”). Screen 3 shows the rest of the conversation (user says “I want to notify Alan” and the agent asks “What message do you want to send your recipient?”). The fourth and last screen of Figure 5 shows the message to be sent (“Please go to the train station as soon as possible.”) and a summary given by the agent to be sure the dialogue has been efficient and the request has been understood. The alert can now be created on the Geohub through the Deveryware geolocation API.

Answers that would be considered as being off topic do not modify the dialogue run because, after their analysis, the validation phase will always point to the next missing parameter so the agent will always ask to fill it out. Besides, the term “son” has been declared in the lexicon and tagged as *MOBILE*; “Alan” has also been declared and tagged as *RECIPIENT*. In its final version, the software agent will be connected to the contacts and profiles of each user. The profiles contain the mobiles declared by the users, the address lists, and the zones and points of interest the users have recorded.

While running a real-case example, we can see that an alert is triggered 24 seconds after the mobile leaves the road (the corridor, actually): indeed, the vehicle left at 10:32:02 and the alert is given at 10:32:26 (see Figure 6).

## 5. Conclusion

Starting from a real-case study for a company, this paper proposes and explains a generic method to take into account natural language in a geolocation interface where an implementation has been proposed. A lexicon has been created, dedicated to the geolocation field using PoS tagging and dictionaries of synonyms. The internal representation of the imprecision, that is, the fuzzy semantics of the sentences,

is dealt with in a model inspired by the 2-tuple fuzzy linguistic representation model by Herrera and Martínez where unbalancement is the key concept. The partition is constructed while taking into account the semantics of the concepts (synonyms and resemblance rate matrices). A dialogue software agent has been implemented on Android platform.

In future works, we will formalize the partitioning method that uses synonyms. We also want to introduce more contextual elements and ontologies to be able to understand more words and to deal with several languages.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

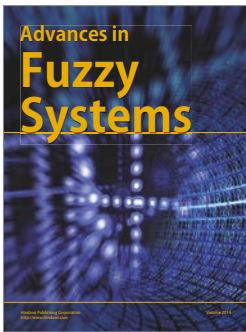
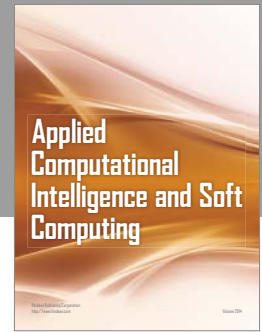
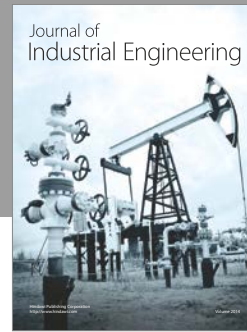
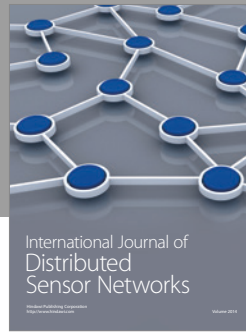
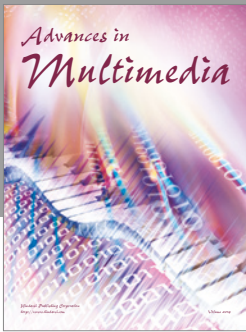
This work is sponsored by the French National Research Agency (ANR) under Grant no. ANR-09-SEGI-012.

## References

- [1] I. Truck and M.-A. Abchir, “Towards a classification of hesitant operators in the 2-tuple linguistic model,” *International Journal of Intelligent Systems*, vol. 29, no. 6, pp. 560–578, 2014, special issue on “Hesitant Fuzzy Sets: An Emerging Tool in Decision Making”.
- [2] F. Herrera, E. Herrera-Viedma, and L. Martínez, “A fuzzy linguistic methodology to deal with unbalanced linguistic term sets,” *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 2, pp. 354–370, 2008.
- [3] N. Chomsky, *Knowledge of Language, Its Nature, Origin, and Use*, Praeger, New York, NY, USA, 1986.
- [4] R. Kuhn and R. D. Mori, “A cache-based natural language model for speech recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 6, pp. 570–583, 1990.
- [5] T. Winograd, “Procedures as a representation for data in a computer program for understanding natural language,” *Cognitive Psychology*, vol. 3, no. 1, pp. 1–191, 1972.
- [6] J. Weizenbaum, “ELIZA—A computer program for the study of natural language communication between man and machine,” *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.
- [7] A. K. Joshi, “Factoring recursion and dependencies: an aspect of tree adjoining grammars (Tag) and a comparison of some formal properties of Tags, GPSGs, Plgs, and LFGS,” in *Proceedings of the 21st Annual Meeting on Association for Computational Linguistics (ACL ’83)*, pp. 7–15, ACL, Stroudsburg, Pa, USA, 1983.
- [8] Z. S. Harris, *A Theory of Language And Information: A Mathematical Approach*, Clarendon Press; Oxford University Press, Oxford, UK, 1991.
- [9] N. Chomsky, *Questions de sémantique*, Edition du Seuil, 1975, translated by B. Cerquiglini.
- [10] A.-M. Popescu, O. Etzioni, and H. Kautz, “Towards a theory of natural language interfaces to databases,” in *Proceedings of the 8th International Conference on Intelligent User Interfaces*, pp. 149–157, ACM, New York, NY, USA, 2003.

- [11] J. Berant, A. Chou, R. Frostig, and P. Liang, "Semantic parsing on freebase from question-answer pairs," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '13)*, vol. 2, 6 pages, 2013.
- [12] L. Shi and R. Mihalcea, "Putting pieces together: combining FrameNet, Verbnet and WordNet for robust semantic parsing," in *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 100–111, Springer, 2005.
- [13] M. Lafourcade and V. Prince, "Synonymies et vecteurs conceptuels," in *Proceedings of the 8ème conférence sur le Traitement Automatique des Langues Naturelles (TALN '01)*, pp. 233–242, 2001.
- [14] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [15] F. Herrera and L. Martínez, "A 2-tuple fuzzy linguistic representation model for computing with words," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 6, pp. 746–752, 2000.
- [16] M. Delgado, F. Herrera, E. Herrera-Viedma, M. J. Martin-Bautista, L. Martínez, and M. A. Vila, "A communication model based on the 2-tuple fuzzy linguistic representation for a distributed intelligent agent system on Internet," *Soft Computing*, vol. 6, no. 5, pp. 320–328, 2002.
- [17] R. Alcalá, J. Alcalá-Fdez, F. Herrera, and J. Otero, "Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation," *International Journal of Approximate Reasoning*, vol. 44, no. 1, pp. 45–64, 2007.
- [18] C. Kahraman, *Fuzzy Applications in Industrial Engineering*, Springer, 2007.
- [19] Y. J. Xu and H. M. Wang, "Approaches based on 2-tuple linguistic power aggregation operators for multiple attribute group decision making under linguistic environment," *Applied Soft Computing Journal*, vol. 11, no. 5, pp. 3988–3997, 2011.
- [20] F. J. Estrella, M. Espinilla, F. Herrera, and L. Martínez, "FLINT-STONES: a fuzzy linguistic decision tools enhancement suite based on the 2-tuple linguistic model and extensions," *Information Sciences*, vol. 280, pp. 152–170, 2014.
- [21] V. C. Gerogiannis, E. Rapti, A. Karageorgos, and P. Fitsilis, "On using fuzzy linguistic 2-tuples for the evaluation of human resource suitability in software development tasks," *Advances in Software Engineering*, vol. 2015, Article ID 695873, 15 pages, 2015.
- [22] L. Martínez, D. Ruan, and F. Herrera, "Computing with words in decision support systems: an overview on models and applications," *International Journal of Computational Intelligence Systems*, vol. 3, no. 4, pp. 382–395, 2010.
- [23] L. Martínez, R. M. Rodríguez, and F. Herrera, *The 2-tuple Linguistic Model—Computing with Words in Decision Making*, Springer, 2016.
- [24] M.-A. Abchir, "A jFuzzyLogic Extension to Deal With Unbalanced Linguistic Term Sets," *Book of Abstracts*, pp. 53–54, 2011.
- [25] M.-A. Abchir and I. Truck, "Towards a new fuzzy linguistic preference modeling approach for geolocation applications," in *Proceedings of the EUROFUSE Workshop on Fuzzy Methods for Knowledge-Based Systems (EUROFUSE '11)*, pp. 413–424.
- [26] M. Espinilla, J. Liu, and L. Martínez, "An extended hierarchical linguistic model for decision-making problems," *Computational Intelligence*, vol. 27, no. 3, pp. 489–512, 2011.
- [27] I. Truck, "Comparison and links between two 2-tuple linguistic models for decision making," in *Knowledge-Based Systems*, vol. 87, pp. 61–68, Elsevier, 2014.
- [28] M.-A. Abchir and I. Truck, "Towards an extension of the 2-tuple linguistic model to deal with unbalanced linguistic term sets," *Kybernetika*, vol. 49, no. 1, pp. 164–180, 2013.
- [29] M.-A. Abchir, I. Truck, and A. Pappa, "Dealing with natural language interfaces in a geolocation context," in *Proceedings of the 10th International FLINS Conference on Computational Intelligence in Decision and Control*, pp. 806–811, Istanbul, Turkey, 2012.
- [30] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, pp. 513–523, 1988.
- [31] B. Green and G. Rubin, "Automated grammatical tagging of english," Tech. Rep., Department of Linguistics, Brown University, 1971.
- [32] S. Ploux and B. Victorri, "Construction d'espaces sémantiques à l'aide de dictionnaires informatisés des synonymes," *Traitement Automatique des Langues*, vol. 39, no. 1, pp. 161–182, 1998.
- [33] J. François, J.-L. Manguin, F. Cordier, and R. Christine, "Le Dictionnaire Electronique des Synonymes du CRISCO: méthode d'utilisation et exploitation psycholinguistique," in *DistanceS*, cahier 2, pp. 165–200, Conseil Québécois de la formation à distance (CQFD), 2004.
- [34] V.-M. Pho, "Génération de réponses pour un système de questions-réponses," in *Proceedings of the Conférence en Recherche d'Informations et Applications (CORIA '12)*, pp. 449–454, Bordeaux, France.
- [35] Z. S. Harris, "Discourse analysis," *Language*, vol. 28, no. 1, pp. 1–30, 1952.





# Hindawi

Submit your manuscripts at  
<https://www.hindawi.com>

