The Language Extended Lexicon, Revisited

Ricardo Wehbe

INTEC – UADE   rwehbe@uade.edu.ar

Abstract. We present in this paper some redefinitions of concepts already well-known in the literature about the Language Extended Lexicon (LEL), prompted by the medium-term objective of constructing a process to assist the user in the construction of LEL. The underlying idea is to look at LEL from a different perspective, rather as a mathematical object than as a Requirements Engineering methodology aimed at deriving scenarios. The main LEL concepts are presented after a review of some of the semiotics concepts that originally inspired it. We outline the main research lines, the problems we expect to encounter, and the approaches, mainly borrowed from the Data Mining area, by which we expect to tackle them.

Keywords: Requirements, LEL, Text Mining

1 Introduction

The Language Extended Lexicon, henceforward LEL [18] was originally developed as a tool to construct scenarios [1]. A scenario is a structured narrative of a situation of the system [11] and it is a partial description since it does not exhaust all possible states of the system. Scenarios consist of two parts, *behaviour* and *situation*. There is thus a natural link between scenarios and LEL [20]. Notwithstanding that, in this work we will let aside scenarios to concentrate in LEL, which deserves study *per se*.

LEL may be concisely described as a glossary with *rôles* and behaviours and it represents a meta-model of the domain. It is an interesting approach that is nevertheless not widely used. The simplistic approach to ascribe this to a lack of knowledge is unconvincing. The reason seems to be the difficulty to construct it. Therefore we thought some kind of computer-assisted tool would be helpful. The medium-term objective we aim at is the construction of LEL in a semi-automatic way starting from requirements documents. A first step in this direction was done in [6]. The approach taken there was a purely syntactic one: relevant words are selected and ranked on the basis of their frequency of occurrence. This approach was based on one of the very first works on the area of automatic abstracting [14], and progress has been swift in this area (see for example [12, 13, 17, 27]), so that there is nowadays enough room for improvement. In order to carry out this project, we need a more precise (in the mathematical sense of the term) description of LEL. This work is a first attempt in this direction.

This paper is organised as follows: Section 2 contains a concise intuitive description of LEL. It is stated in [18] that the roots of LEL are on Semiotics rather than on Computer Science, and Section 3 is a brief exposition of some relevant ideas of Semiotics which will be used in Section 4, where we rewrite some definitions of LEL to better serve our purposes. Finally, in Section 5 we discuss some approaches to set about some of the problems that remain open.

2 An Informal Description of LEL

The description of LEL we outline here is a very succinct one, just enough for the understanding of the following sections. We will paint with rather broad brush here. The reader is referred for more details to Graciela Hadad's PhD thesis [11], which contains a good summary of LEL.

The main assumption underlying LEL is that the involvement and compromise of users and customers increases when they share a common language with the software engineers [15]. It is thus important that the latter capture as much jargon of the former as possible. LEL is a methodology to achieve this and the interesting part is that, during the LEL construction phase, the emphasis is on the vocabulary rather than on the problem [18]. This does not mean that the problem is underestimated; it only means that during the construction phase of LEL, it is the glossary that counts.

A central concept of LEL is that of *Universe of Discourse*. The Universe of Discourse is defined in [18] as "the context set by the systems engineering process" where the "systems engineering process defines the context and the goals of the software artefact."

Each symbol is associated with a *notion*, stating what the symbol *is* and with an *impact* stating what the symbol *does*. We will come back on this later, but we can identify the notion of a symbol with its meaning and the impact of a symbol with its behaviour. There is also, besides the LEL symbols, a *minimal vocabulary* containing some symbols of common use in the natural language not belonging to LEL, such as articles and some common verbs, adjectives, and nouns [11].

The syntax of LEL is summed up by the grammar of Fig. 1, adapted from [11].

⟨LEL⟩	→	⟨Symbol⟩
⟨Symbol⟩	→	⟨Symbol⟩⟨Symbol⟩ | ⟨Name⟩⟨Notion⟩⟨Impact⟩
⟨Name⟩	→	**Name:**⟨DefName⟩
⟨DefName⟩	→	⟨Word⟩ | ⟨Phrase⟩ | ⟨Acronym⟩
⟨Notion⟩	→	**Notion:**⟨DefNotion⟩
⟨DefNotion⟩	→	⟨Statement⟩
⟨Impact⟩	→	**Impact:**⟨DefImpact⟩
⟨DefImpact⟩	→	⟨Statement⟩

where word, phrase, acronym, and statement have the usual meanings. They are constructed either with symbols of LEL or with symbols of the minimal vocabulary.

Fig. 1. The syntax of LEL

Let us consider the example of a LEL symbol shown in Fig. 2, taken from [19]:

> **Name:** Caller
> **Notion:**
> – person who calls the participants to a meeting
> – may be a participant
> **Impact:**
> – determines the meeting's objectives and its agenda
> – determines the invitees and the material to be distributed
> – makes the design of meetings agenda
> . . . .

Fig. 2. A symbol of LEL: the caller to a meeting

Sometimes the impacts are called *behavioural responses* [5, 18]. The example of Fig. 2 corresponds to the symbol of the caller to a meeting. Observe that there are several words that are underlined. These are the *relevant* symbols and all these symbols of the Universe of Discourse should have a LEL entry. Relevant symbols should be defined in terms of other relevant symbols or in terms of the symbols of the minimal vocabulary.

Observe also that not all the underlined words have the same *rôle* if we consider the intuitive meaning: there are persons (caller, invitee), there are objects (objectives of the meeting), and there are actions (design of the meetings agenda.) All these corresponds to different types of symbols [11]:

- Subject: an active entity, which performs activities in the application's domain.
- Object: a passive entity, to which actions may be applied, but which does not perform any action.
- Verb: an activity or action in the application's domain.
- State: a condition or situation in which subjects, objects or verbs of the application's domain may find themselves at a given time.

Observe that each one of the preceding types will have an entry similar to that of Fig. 2. The construction of LEL is guided by two principles: the minimisation of the usage of the symbols external to the LEL and the maximisation of the usage of the symbols belonging to the LEL. The latter is sometimes identified with a *circularity principle* [5], although it is not clear that both concepts are necessarily equivalent. We understand under circularity the need to have an entry for every symbol relevant to the LEL: in other words, the constraint already mentioned that every symbol of LEL be defined in terms of symbols of LEL or of symbols of the minimal vocabulary.

This is graphically depicted in Fig. 3, where black circles represent symbols of the minimal vocabulary, white circles symbols of the LEL, and an arrow from a symbol A to a symbol B means that B is defined in terms of A.

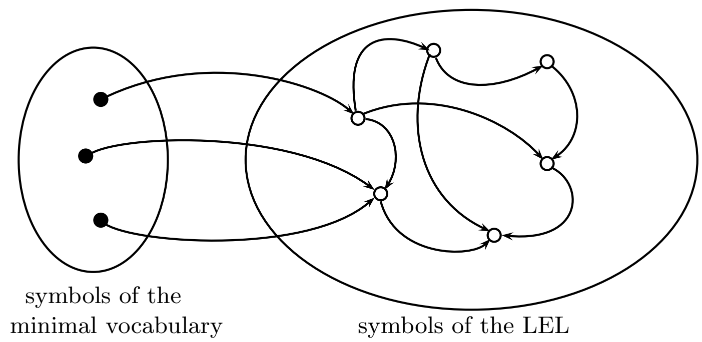

Fig. 3. An example of a circular definition of symbols

We will come back to this circular definition of symbols in Section 4, which entails some interesting problems.

One important point is whether it is possible to automatically classify a word into one of the four types of LEL symbols. As a first approach, nouns should be either classified as subjects or objects, verbs should naturally be classified as verbs and adjectives as states. This is not as straightforward, as the example depicted in Fig. 4 shows [6].

Design of Meetings Agenda
Notion:
- activity carried out by the caller to determine the date, time and place of meetings based on the available time of the invitees to a meeting according to space availability
- aims at organising time and avoiding overlaps of meetings

Impact
The date, time, place of the meeting is registered in its agenda
....

Fig. 4. Another symbol of LEL: the design of meetings' agendas.

This symbol is correctly classified as type "verb", although the Spanish word used in the original ("diseño") corresponds to a noun. We see here where the technical problem lies: it is difficult to assign a type to a symbol without some semantic knowledge, and this introduces complications. There are other points: for instance, the difference between “subject” and “object” is a purely semantic one, since it relies on the fact that the denotation of the symbol be an active or passive entity.

Before closing this section, we make some observations:

- The construction of the list of symbols of LEL typically proceeds in an incremental way: starting from users’ documents, interviews, and general domain knowledge, a list of symbols is initially created. The symbols of this list require other symbols for their definition, which in turn become part of the list. The process ends when no new symbols enter the list.
- The process above makes no difference among subjects, objects, verbs and states.
- All symbols are exclusively taken either from the Universe of Discourse or from the minimal vocabulary $\mathcal{M}$.

3 Intermezzo: A Short Digression on Semiotics

Since the founding concepts of the LEL approach are taken from semiotics [18], let us first cast a glance on this discipline. It is indeed a very interesting branch of knowledge with some deep relations to computer science. We will point out some of these relations as they turn up.

The basic assumption is that all forms of communication function as the emission of messages based on underlying codes [8]. The reader may recall the famous communication model of Shannon [24], which is reproduced in Fig. 5.

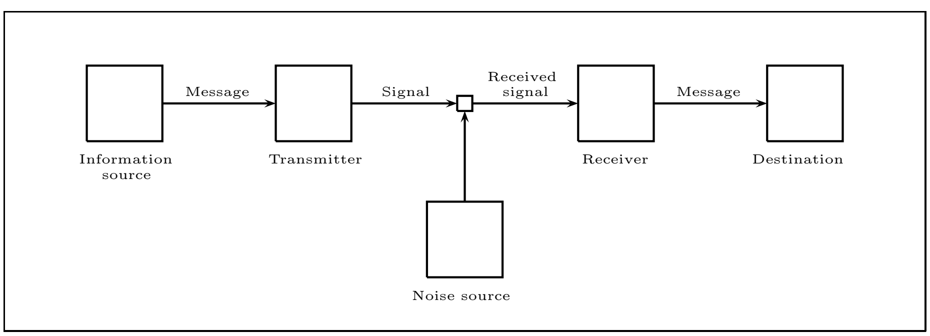

Fig. 5. The abstract communication model of Shannon

Semiotics makes sense if we accept the need of some convention which makes the message intelligible for the receiver. The duality between the message and its meaning is nothing else but the duality between syntax and semantics. This is especially clear already in the first works on semiotics [23]. Of course, the semiotic field is much more complex than the realm of formal languages familiar to computer scientists. Semiotics aims at a *structure* that may turn the chaotic complexity of the semiotic field into a *system* [8]. In a similar way, LEL aims at giving the disorder prevailing in requirements documents, meeting minutes, and transcripts of interviews a structure to turn them eventually into a set of scenarios.

A central conception of semiotics is the difference among *denotation* and *connotation*. This is not a new feature of Semiotics: the need for both concepts can at least be traced back to the 19th century, for instance the distinction between *Sinn* (sense) and *Bedeutung* (meaning) in Frege [10]. Denotation is relatively straightforward: it is the meaning of a symbol. In contrast, connotation is a more subtle concept. Consider the following example [9]: there is a hydraulic system and there is a control office where a technician may receive the following messages: 00, 01, 10, and 11. The messages have the following denotations:

Table 1. Denotations of signals in a hypothetical hydraulic system.

expression	meaning
00	normal
01	low water
10	alarm
11	danger

But when the technician receives a message, he has to act in response to it. If we take the response into account, we get something like the following table:

Table 2. A more complex relationship between denotations and connotations.

expression		meaning
expression	meaning	
00	normal	relax
01	low water	open bomb
10	alarm	give alarm
11	danger	evacuate

The response in the third column has not been communicated by the message, but has been *signified* by it [9]: we have a meaning communicated by a *preceding meaning*. A semiotic that is constituted by another one at the level of the message is *connotative*. In the preceding example, the expression 11 taken together with its denotation (danger) becomes the expression of an ulterior meaning: the message 11 *denotes* danger and *connotes* evacuation.

Such a situation implies superposition of semiotic planes[1] that makes things more complex. In particular, our communicational system with natural language is a complex connotative network. Fortunately, our Universe of Discourse is not that complicated. Not only the denotations but also the connotations can be expressed within it.

Some observations before going back to LEL.

- It is essentially correct to identify the concept of *notion* as defined in LEL with that of *denotation* of Semiotics.
- As soon as we have a semiotic field of relative low complexity, we may identify the concept of *impact* or *behavioural response* of LEL with that of *connotation* of Semiotics.

4 Back to LEL

The Universe of Discourse as defined in Section 2 turns out to be rather elusive stuff, no matter how clear its intuitive meaning might be. It is difficult to define such a concept precisely, since it involves a heterogeneous collection of documents, scripts of interviews, informal communications, and so on. On top of this, sometimes the whole information gathered turns out to be inconsistent.

We will need some definitions before going on.

Definition 1 (Core vocabulary). *Given a set of words $\mathcal{V}$, the* core vocabulary *of $\mathcal{V}$, denoted by $\overline{\mathcal{V}}$, is the set formed by subsuming any word $w \in \mathcal{V}$ onto one* representative *$w' \in \overline{\mathcal{V}}$ such that all distinctions among morphological variants of w are eliminated in $\overline{\mathcal{V}}$.*

For instance the representative of a noun is its singular form and the representative of a conjugated verb is its infinitive form. For instance, the representative of *tables* and *table* is *table*; the representative of *been* and *was* is *to be*. In a way, the core vocabulary is the list of words as it would appear in a dictionary. We assume from now on that the minimal vocabulary $\mathcal{M}$ (or equivalently its core version $\overline{\mathcal{M}}$) are specific of the *language* and not of the *domain of application*: there is a core vocabulary for Spanish, one for English, and one for German; there is not one for Accounting and another for Databases. We may thus assume that $\mathcal{M}$ (and therefore $\overline{\mathcal{M}}$) is uniquely determined once the language has been fixed.

Definition 2 (Universe of Discourse (UoD)). *Let $\mathcal{D}$ be the set containing all words belonging to the whole bunch of documentation collected during the requirements process. The set* $\mathsf{UoD} = \overline{\mathcal{D}} \setminus \overline{\mathcal{M}}$ *is the* universe of discourse *derived from $\mathcal{D}$.*

[1] In the sense that a signal denotes a meaning and this meaning in turn denotes another one.

Observe that it is not necessarily the case that all words appearing in UoD must occur in the requirements documentation, but all words appearing in UoD are representatives of some word occurring in that documentation.

The first technical problem is to assign to each one of the words in UoD a type. This is difficult because it requires some semantic understanding. For instance, the difference between *subject* and *object* lies in the activity or passivity of the element being observed. But to establish that it is necessary to know the *rôle* played by the element denoted by the word and it is very difficult to do that with purely syntactic tools. Another problem lies in the words of type *verb* which are not necessarily described by a verb (recall the example of Fig. 4.) We will discuss some possible ways to tackle this problem in Section 5. For the time being, we will assume that we have assigned each word a classification, either by manual work or by some kind of magic[2].

Summing up, we have a first stage of *collection* of words that ends with a classification of the words collected in UoD, where each one is assigned a type. The SADT[3] diagram shown in Fig. 6, which is a simplified version of the one in [11], depicts this part of the process:

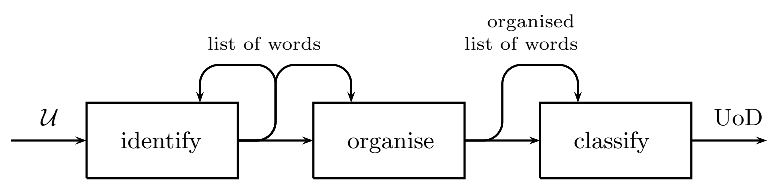

Fig. 6. First stage of the construction of LEL: collection of symbols

Recall that in a SADT diagram, ingoing arrows from the left are inputs, ingoing arrows from above are tools and outgoing arrows (only rightwards) are outputs [21]. As commented above, the critical part to do this semi-automatically is the last step, the classification of the symbols according to their type.

After the collection phase, the *description* phase ensues. Here each word is assigned a notion and an impact. In other parts, the LEL symbols are assembled. It is not realistic to expect that this part of the process be carried out in a fully automatic way, since it requires a deep semantic understanding. Again, we discuss some possible approaches to solve this problem in Section 5.

So far, we have obtained a set of LEL symbols, which are generated by the grammar of Fig. 1 where for each of word occurring in the non-terminals $\langle\mathsf{word}\rangle$, $\langle\mathsf{phrase}\rangle$, $\langle\mathsf{acronym}\rangle$, and $\langle\mathsf{statement}\rangle$ there is a representative in $\overline{\mathcal{M}} \cup \mathsf{UoD}$.

[2] "The method of 'postulating' what we want has many advantages; they are the same as the advantages of theft over honest toil." Bertrand Russell [22], p. 71.

[3] The acronym stands for *Structured Analysis and Design Technique*.

Another problem that must be undertaken is the completeness of the model. There are two types of completeness, internal and external [7]. The former refers to completeness within the symbol in the sense that each symbol has been completely described and the latter refers to completeness outside the symbol, in the sense that all possible symbols have been identified. Needless to say, verification of any of these types of completeness is hard.

We introduce yet more notation. We denote by lower-case Greek letters the symbols of the LEL and by Λ the set of all LEL symbols. Further, given a LEL symbol $\varphi \in \Lambda$, we denote by $\mathsf{name}(\varphi)$ the name of it, by $\mathsf{notion}(\varphi)$ the set of words of UoD occurring in its definition of notion, and by $\mathsf{impact}(\varphi)$ the set of words of UoD occurring in its definition of impact. In other words, we have:

$$\begin{aligned} \mathsf{name} &: \Lambda \to \mathsf{Uod} \\ \mathsf{notion} &: \Lambda \to \wp(\mathsf{Uod}) \\ \mathsf{impact} &: \Lambda \to \wp(\mathsf{Uod}) \end{aligned}$$

Where $\wp(S)$ denotes the power set of a set S. Observe that, for a given LEL symbol φ, its notion and impact contain words of both sets UoD and $\overline{\mathcal{M}}$. But for the definition of $\mathsf{notion}(\varphi)$ and $\mathsf{impact}(\varphi)$ we are only interested in the symbols of UoD. We define now the partial function $\mathsf{symbol} : \mathsf{UoD} \to \Lambda$.

$$\mathsf{symbol}(w) = \begin{cases} \varphi \in \Lambda & \text{if } \mathsf{name}(\varphi) = w \\ \text{undefined} & \text{otherwise} \end{cases}$$

Observe that if Λ is externally complete (Def. 4), then symbol is no longer a partial function but a normal function, since it is defined in all its range.

Our approach to completeness will be based on Data Mining and Machine Learning concepts. Next we recall the notion of distance function between elements of an arbitrary set S.

Definition 3 (Distance Function). *[16] Let S be a set and let $x_1, x_2 \in S$. Then $\mathsf{d} : S \times S \to \mathbb{R}$ is a* distance function *within S if and only if the following conditions are satisfied for any $x_1, x_2, x_3 \in S$:*

1. $\mathsf{d}(x_1, x_2) \geq 0$ *with* $\mathsf{d}(x_1, x_2) = 0$ *only when* $x_1 = x_2$ *(positiveness.)*
2. $\mathsf{d}(x_1, x_2) = \mathsf{d}(x_2, x_1)$ *(symmetry.)*
3. $\mathsf{d}(x_1, x_2) \leq \mathsf{d}(x_1, x_3) + \mathsf{d}(x_3, x_2)$ *(triangle's inequality.)*

We will assume the existence of some distance measure in UoD (i.e., some distance measure between words.) We put off a detailed discussion to Section 5.

Definition 4 (Completeness). *Let Λ be a set of LEL symbols, let $\varphi \in \Lambda$ be a LEL symbol. Let further $\mathsf{d} : \mathsf{UoD} \times \mathsf{UoD} \to \mathbb{R}$ be a distance function in UoD and let $k \in \mathbb{R}$. We say that φ is an* internally complete *symbol with respect to k and d if all words $w \in \mathsf{UoD}$, it is the case that*

$$\mathsf{d}(w, \mathsf{name}(\varphi)) \leq k \textit{ implies } w \in \mathsf{notion}(\varphi) \cup \mathsf{impact}(\varphi)$$

We say that φ *is a* minimal internally complete *symbol with respect to* k *and* d *if all words* $w \in \mathsf{UoD}$*, it is the case that it is an internally complete symbol with respect to* k *and* d *and it is the case that*

$$w \in \mathsf{notion}(\varphi) \cup \mathsf{impact}(\varphi) \textit{ implies } \mathsf{d}(w, \mathsf{name}(\varphi)) \leq k$$

We say that Λ *is* externally complete *if it is the case that*

$$\textit{For all words } w \in \mathsf{UoD} \textit{ there is a } \varphi \in \Lambda \textit{ such that } w = \mathsf{name}(\varphi)$$

We are ready to show that it is straightforward to achieve external completeness, which is to be identified with the circularity or closure principle of [5, 11, 18, 19]. First we consider the function $\mathsf{f} : \mathsf{UoD} \rightarrow \wp(\mathsf{UoD})$ defined as follows:

$$\mathsf{f}(w) = \mathsf{notion}(\mathsf{symbol}(w)) \cup \mathsf{impact}(\mathsf{symbol}(w))$$

Now let the function $\mathsf{F} : \wp(\mathsf{UoD}) \rightarrow \wp(\mathsf{UoD})$ be a generalisation of f to subsets of UoD, such that for $V \subseteq \mathsf{UoD}$ we have:

$$\mathsf{F}(V) = \{w \in \mathsf{UoD} \mid w \in \mathsf{f}(v) \text{ for some } v \in V\}$$

Observe that the set $\wp(\mathsf{UoD})$ with the partial order relation $\subseteq$ (set inclusion) is a complete lattice. Observe besides that the F function is monotonic, namely

$$V \subseteq W \text{ implies } F(V) \subseteq F(W)$$

Thus, the conditions of the Knaster-Tarski fixed point theorem [4, 25] are fulfilled and there exists a greatest fixed point, namely there is a largest subset $V \subset W$ such that

$$F(V) = V$$

This set is UoD itself, which is equivalent to saying that the circularity principle is fulfilled. Furthermore, since the set UoD is finite, this fixed point is iteratively reachable in a finite number of steps.

We assume that both denotations and connotations may be defined with the vocabulary of $\mathcal{D}$, whose representatives are in $\overline{\mathcal{M}} \cup \mathsf{UoD}$. This may be a simplistic assumption, but we must recall that we are dealing here with relatively straightforward documents and not with complex literary texts fraught with metaphors and obscure allusions.

5 Conclusions and Future Work

We think that LEL is a valuable tool in the context of Requirements Engineering (and outside it.) Its use is nevertheless not widespread. This is not only attributable to a lack of knowledge of it, but to a greater extent to the difficulty to construct it. A semi-automatic tool to construct LEL would thus be useful.

The first step to achieve this goal is to get a definition of LEL more fitted to our purposes than the ones currently available. There are some remaining problems, as we have already commented in Section 4, which we intend to approach. We mention them in order of appearance.

The first of them is to get a classification for each word of UoD. Each symbol of LEL is classified either as subject, object, verb, or state. It is difficult to perform this task with a purely syntactic approach. Nevertheless, we want to stick to syntactic methods such as those used in Text Mining [26] for as long as possible. The first idea that comes to mind is to use some of the classifying methods already well studied in Data Mining [3, 16], such as Bayes Naïve Algorithm or Clustering. In the first case, a large database is needed to enable the conditional probabilities of a word belonging to a category with a certain degree of confidence. The underlying hypothesis is that the words used in different universes of discourse in Requirements Engineering tend to reappear. This requires a notion of distance between words, discussed next.

The second problem is how to define the denotation and connotation of a symbol, namely its *notion* and *impact*. We do not think this part of the process may be fully automatised, but it is possible that the system propose, given a symbol φ, some candidates a participate in $\mathsf{notion}(\varphi)$ and $\mathsf{impact}(\varphi)$. The first approach we will attempt is inspired in Machine Learning methods, such as k nearest neighbours or clustering [3, 16]. For such method we need to define a distance. We want to begin with a syntactic approach by defining the distance within statements, paragraphs, and pages. Eventually this measure could be enhanced with some semantic features of the word being considered, as in the *Google Distance*[2].

Acknowledgements. The author is grateful to two anonymous referees for their insightful suggestions.

References

1. Benner, Kevin; Feather, Martin; Johnson, Lewis; Zorman, Lorna: *Utilizing Scenarios in the Software Development Process*, Information System Development Process, IFIP Transactions A-30, Elsevier Science Publishers, 1993, pp. 117–134.
2. Cilibrasi, Rudi; Vitányi, Paul: *The Google Similarity Distance*, IEEE Trans. on Knowledge and Data Enginering 19(3), 2007, pp. 370–383.
3. Cios, Krzysztof; Pedrycz, Witold; Swiniarski, Roman; Kurgan, Łukasz: *Data Mining: A Knowledge Discovery Approach*, Springer, 2007.
4. Cousot, Patrick; Cousot, Radhia: *Constructive Versions of Tarski's Fixed Point Theorem*, Pacific Journal of Mathematics 82(1), 1979, pp. 43–57.
5. Cysneiros, Luiz; Sampaio do Prado Leite, Julio: *Using the Language Extended Lexicon to Support Non-Functional Requirements Elicitation*, Proc. of the Workshop on Requirements Engineering (WER), 2001, pp. 139–153.
6. Demitrio, Daniel: *Framework para Elicitación Automática de Conocimientos*. MSc thesis, Universidad Nacional de La Plata, 2005 (in Spanish.)
7. Doorn, Jorge; Ridao, Marcela: *Completitud de Glosarios: un Estudio Experimental*, Proc. of the Workshop on Requirements Engineering (WER), 2003, pp. 317–328.

8. Eco, Umberto: *La estructura ausente. Introducción a la semiótica*, Lumen 1999 (in Spanish.)
9. Eco, Umberto: *Tratado de Semiótica General*, Lumen, 2000 (in Spanish.)
10. Frege, Gottlob: *Über Sinn und Bedeutung*, Zeitschrift für Philosophie und philosophische Kritik 100, 1892, pp. 25–50 (in German.)
11. Hadad, Graciela: *Uso de Escenarios en la Derivación de Software*, PhD thesis, Universidad Nacional de La Plata, 2006 (in Spanish.)
12. Kupiec, Julian; Pedersen, Jan; Chen, Francine: *A Trainable Document Summarizer*, Proc. of the Conference on Research and Development in Information Retrieval (SIGIR), 1995, pp. 68–73.
13. Liang, Shao-Fen; Devlin, Siobhan; Tait, John: *Can Automatic Abstracting Improve on Current Extracting Techniques in Aiding Users to Judge the Relevance of Pages in Search Engine Results?* , Proc. of the 7th Computational Linguistics UK, 2004, pp. 154–159.
14. Luhn, Hans Peter: *The Automatic Creation of Literature Abstracts*, IBM Journal of Research 2, 1958, pp. 159–165.
15. Macaulay, Linda: *Requirements Capture as a Cooperative Activity*, Proc. of the 2nd IEEE Int. Symposium on Requirements Engineering (RE'93), IEEE Computer Society Press, 1993, pp. 174–181.
16. Maimon, Oded; Rokach, Lior (eds.): *Data Mining and Knowledge Discovery Handbook*, Springer, 2005.
17. Prasad, Rajesh; Kulkarni, Uday; Prasad, Jayashree: *Connectionist Approach to Generic Text Summarization*, World Academy of Science, Engineering and Technology 55, 2009, pp. 365–369.
18. Sampaio do Prado Leite, Julio; Franco, Ana Paula: *A Strategy for Conceptual Model Acquisition*, Proc. of the 1st IEEE Int. Symp. on Requirements Engineering (RE'92), IEEE Computer Society Press, 1992, pp. 243–246.
19. Sampaio do Prado Leite, Julio: *Eliciting Requirements Using a Natural Language Based Approach: The Case of the Meeting Scheduler*, Monografía em Ciência da Computação 13/93, Computer Science Department, PUC, Rio de Janeiro, 1993.
20. Sampaio do Prado Leite, Julio; Rossi, Gustavo; Balaguer, Federico; Maiorana, Vanesa; Kaplan, Gladys; Hadad, Graciela; Oliveros, Alejandro: *Enhancing a Requirements Baseline with Scenarios*, Requirements Engineering Journal, 2(4), 1997, pp. 184-198.
21. Ross, Douglas: *Structured Analysis: A Language for Communicating Ideas*, IEEE Trans. on Software Engineering 3(1), special issue on Requirements Analysis, 1977, pp. 16–34.
22. Russell, Bertrand: *Introduction to Mathematical Philosophy*, Allen & Unwin, 1919.
23. Saussure, Ferdinand de: *Cours de Linguistique Générale*, Payot, 1995 (in French.)
24. Shannon, Claude: *A Mathematical Theory of Communication*, The Bell System Technical Journal, 27, 1948, pp. 379-423 and 623-656.
25. Tarski, Alfred: *A Lattice-Theoretical Fixpoint Theorem and its Applications*, Pacific Journal of Mathematics 5, 1955, pp. 285–309.
26. Weiss, Sholom; Indurkhya, Nitin; Zhang, Tong; Damerau, Fred: *Text Mining – Predictive Methods for Analyzing Unstructured Information*, Springer, 2005.
27. Wu, Xiaofeng; Zong, Chengqing: *A New Approach to Automatic Document Summarization*, Proc. of the Int. Joint Conference on Natural Language Processing (IJCNLP), 2008, pp. 126–132.