Which Semantics for Requirements Engineering: from Shallow to Deep

Roberto Garigliano
SenseGraph Ltd, United Kingdom
roberto_garigliano@hotmail.com

Dominic Perini
SenseGraph Ltd, United Kingdom
dominic.perini@gmail.com

Luisa Mich
University of Trento, Italy
luisa.mich@unitn.it

Abstract

Natural language processing has been proposed and applied to support a variety of tasks in requirements engineering. While shallow semantic allows to address many of the challenges, to further automatize requirements analysis a full understanding of textual requirements is needed. To this end, the future generation of natural language processing systems needs a deep semantics, that is a representation of the content independent of the surface description, which represents hidden casual, spatial, temporal and modal connections.

1 Introduction

Natural language processing (NLP) tools and systems have been applied to analysing requirements texts since the 1980's [Ab83]. In software engineering the goal was to design programs by informal English descriptions. Since then, various problems have shown the limits of existing technologies in reaching such objective.

A following wave of papers is related to the arrival of object-oriented methods, suggesting the application of linguistic rules to extract classes and objects from natural language problem statements, in order to develop conceptual models of requirements [Boo84; Rum91; Bur95].

Another area of application of linguistic tools in requirements engineering (RE) is related to the identification of ambiguities in natural language requirements, to improve their quality [Kiy08; Tjo13; Fer16].

More recent research projects focus on the extraction of requirements specifications from regulatory documents, to design computer-based systems compliant with security and privacy laws (e.g., [Gov09]), a critical challenge in an Internet centred world. There are also proposals to analyse user generated content in order to extract requirements (e.g., [Bäu17]) for the purpose of improving products or services as part of a management strategy exploiting textual reviews available on the Web.

Based on the experience gained in more than 30 years, the paper summarises the most relevant mistaken assumptions as regards NLP in RE and illustrates the need of a NLP system able to fully understand the meaning of natural language requirements showing the need for a deep semantics. The main assumptions for the proposed deep semantics are given and an example of its implementation in a large domain independent NLP system, SenseGraph, are described.

2 Mistaken assumptions in NLP and RE

Among the large number of papers published on NLP in RE, it is worth citing [Rya93], who back in 1993 advised "that potential role of natural language processing in the requirements engineering process has been overstated in the past, possibly due to fundamental misunderstandings of the requirements engineering process itself." The author proposed to identify activities where NLP could be usefully applied, underlining the difficulty in taking into account common knowledge, that is knowledge that is not (and could not be) explicitly given in requirements documents (e.g., who a customer or a user is), but also the difficulties inherent to the requirements engineering for complex systems. In this way, Ryan suggested to take into account the state of the art of linguistic tools available at that time and the need to further investigate how to exploit NLP in RE.

Burg [Bur97] introduced a RE method which makes heavy use of linguistic instruments, suggesting the use of semantics to avoid ambiguity and incompleteness problems in natural language texts. However, his approach deals with the semantics at the level of single words, whose meaning is described using a formal representation.

Many other projects have focused on the use of linguistic tools and systems in RE. An analysis of the literature highlights two relevant and somehow opposite attitudes:

- *The under-estimation of the complexity of natural language*, which introduces a number of limitations on the input or grammar or domain etc.
- *A focus on very specific issues or tasks* such as, for example, identifying entities, solving anaphorical references, looking for a given type of ambiguity.

Both those trends confirm the need of a new generation of NLP systems, able to deal with meaning in a way closer to that understood by native users.

From our first project with the NLP LOLITA, to support the generation of class and use cases models three issues – mistaken assumptions – were identified [Mic94; Mic96; Mic02]:

- *The isomorphism between syntax and semantic, that is between the role of the words in the sentences and their meaning*: it is imperfect, so that, for example, rules to extract objects looking for nouns and verbs for methods may not work (nouns can be *verbalised* and verbs can be *nominalised* [Boo94]).
- *All the information is in the text*, but it is not, as for example common knowledge is given for granted [Len90].
- *All the information in the text is useful*, that is not true as there could be redundancies, or misinterpreted facts or ambiguities, etc. (e.g., [Gén13]).

Experiments run with LOLITA – a large NLP system designed according to the principle of natural language engineering [Bog95] – to identify ambiguities in natural language requirements allowed to introduce measures for different kind of ambiguities, but not to support their elimination [Mic00].

For some these assumptions could be considered simplifying assumptions and they are if they are made consciously. If the decision of using the simplifying assumptions is taken by a human who has read the documents and who understands the client's needs, then of course it is fine, but it defeats the objectives of automatic processing. If it is done by machines, it is highly dangerous, as the parts left out might be crucial; e.g., let's assume that the requirement is "the system will be kept going at all times, provided the temperature does not rise over X". If the "provided" part is eliminated, the precision is still 100%, but the result could be disastrous.

On the other hand, the need of tools supporting requirements engineers in dealing with real natural language input – and not assuming that it would be possible to have the requirements in formal language or written in a controlled language – has been confirmed by a survey run in 1999 [Mic04]. At the same time, the state of the art of NLP tools and their limitations prompted to identify tasks in RE that could be addressed using lightweight linguistic tools for semi-automatic approaches in which manual activities are necessary to complete or supervise the tasks [Kiy08; Zen07]. Finally, the development of systems to analyse regulatory texts to extract compliance requirements [Zen15; Zen17] further confirmed that supporting RE to a better extent needs a full-fledged NLP system.

This is the goal of SenseGraph, i.e. to implement a deep semantic analysis which is close to what some cognitive science research indicates as a human internal representation [Joh06]. In our vision, SenseGraph can be feed with NL requirements documents with minimal human editing (e.g., to eliminate pictures, HTML tags, etc.) and extract a deep semantic version which clearly shows the crucial causal and temporal links, and which could be queried by NLP too. In this way requirements engineering activities that now have to be supervised and completed by humans could be further automatised, overcoming the limitations of linguistic analysis based on shallow semantics.

3 Deep semantics

Natural language texts are usually analyzed in a sequential process, starting with lexical and structural elements, parsing text to identify the most suitable parsing tree and then applying more or less complex techniques to interpret the semantic

content that is to understand the meaning. This sequence of analysis does not allow to understand fully the content nor to obtain a representation of the meaning independent on the surface description.

This paper proposes a deep internal semantic representation of the text, which attempts to describe its meaning in a form that can be (even very) different from the original one and to extract information that is not explicitly given in the text.

The deep semantics provided by SenseGraph (http://www.sensegraph.com) is developed following a strongly minimalist theory:

- The basic event representation units required by the model are the simple subject-action-object or subject-action (for intransitive actions). These units take place in time-space, thus they needed to be positioned either in 'absolute' terms (e.g. an address such as via Garibaldi 31, Messina, or a specific time such as 10th January 2018) or in relative ones. More importantly from the point of view of discourse analysis, such basic units are also positioned with respect to one another, e.g. for space correlations (event A taking place 50 meters from Event B) or time correlations (event A takes place before event B). Of course these are 'absolute' or 'relative' only in natural language terms, not in any scientific meaning such as in relativity theory.
- Causal relationships are a subset of the temporal ones, which have the specific feature of forcing some sequence aspects (e.g. necessity, sufficiency etc.). The only other relationships between events are those that create the transition from a common world to a personal one. For example, in "Tom thinks that Mary is pretty", the fact that "Tom thinks …" is in the common world, while the fact that "she is pretty" is true in Tom's world. A similar thing happens in the relationship between message and content (e.g. "the article contains the story of the robbery").
- All other linguistic expressions must be reduced to this simple model, without losing any meaning understandable by a competent reader from the surface structure. Given the huge variety of surface forms, and the very limited set of primitives in the deep semantic, this transition requires a set of rules which are very complex, both as theory and as implementation. Most of the theory has been developed and a good deal of it already implemented.

SenseGraph has been used, and preliminarily validated, in a national project and in an European project. In these projects, the system was tasked with analysing texts from similar domains (terrorism for the national project and crime for the European project), while the type of text was different (short, information rich Reuters flash news in Syntesis; long newspaper articles and blogs in LASIE). In both cases, the goal was to produce an analysis which helped investigators to provide a clear representation of the information and the underlying structures. In order to reach these objectives, the deep semantic representation has proved the key feature, since it has allowed to unify apparently different entities and events and to connect them using implicit deep temporal, causal and spatial chains. It has also been essential in extracting motivations, likely actions, elements of planning and other mental structures.

Deep semantics is particularly suited to requirement analysis, since it leads to very standardised representations from texts which may appear very different on the surface. It also allows easy and efficient reasoning, because there are so few types of links allowed between events. The following example illustrates as the original text or its surface representation - parsing or semantic - is very distant from its deep semantics. The input is the following sentence, which could be part of a text used to create a database to store data about crimes for the police:

(a) A 59-year-old man from York has been arrested on suspicion of murdering missing chef Claudia Lawrence.

A shallow semantic analysis of the sentence cannot help in answering questions as e.g., "Who arrested the man?", or even more complex one such as "Why was the man arrested?", nor could the parsing tree help more, because of its dependency on the surface structure. In this sentence a lot of knowledge is implicit, but a reader would be able to interpret it, understanding it as follows:

Claudia Lawrence worked as a chef, then she disappeared, then she may have been murdered, then police suspected that a man murdered her and so they arrested him. He had been in York before police arrested him, and was 59 years of age when the police arrested him. Thus, deep semantics means that all the implicit information (e.g. events hidden inside nouns such as suspicion, adjective such as missing or roles such as chef) has to be extracted and organised in small atomic unit, which then are put together in the correct temporal and causal sequence.

In SenseGraph, information is represented as objects and events, and for that sentence the system creates 4 objects and 14 events. The 4 objects are the concept of man, York (used in the event which describes the man's position before the arrest), Claudia Lawrence, and police, derived from the subject of the general event used to represent an arrest. Arrest is an example of a general event marked as prototypical, which allows to explicit police as the subject of the arrest. Other prototypes used to represent the meaning of the sentence are that of murder and suspicion. Police is also used as the subject of the events "Police arrests a man", "Police suspects a man murdering Claudia Lawrence", "Police suspects a man murdering Claudia Lawrence so they arrests him". The last event represents the reason of the arrest, i.e. the causal link between suspicion and arrest.

The meaning of the last part of the sentence is represented by the following events: “Claudia Lawrence is a chef”, “Claudia Lawrence disappears”, and “Claudia Lawrence works as Chef, before she disappears”. The murder, the suspicion and the arrest are connected by the event “Police suspects a man murdering Claudia Lawrence so they arrests him.” The other events are needed to represent the causal, spatial and temporal relations among the events in the original sentence. Notice how the suspicion is real in the police ‘world’ (at least enough to cause the arrest), but only ‘hypothetical’ in the reporter’s world.

The following phrase:

(b) Police have arrested a York man, aged 59, because they suspect him to be the murderer of Claudia Lawrence, the chef who has disappeared.

has the same meaning for any competent native speaker as phrase (a), but it produces a completely different parse tree and surface semantics. Our system, however, produces the same deep representation. It should be noticed that there is a large amount of ways in which this same meaning could be expressed.

Figure 1 illustrates an extract of the output produced by SenseGraph for the sentence: the event arrest, the object police and the event created to explicit the fact that the man is 59 years old when he is arrested by the police.

Notice how the final analysis is rather distant from the original text, although according to the Mental Models Theory is very close to how a native speaker would mentally visualize the story [Joh06]. The system presents this information in an interactive graphical form, as well as in a textual one.

*** arrest/1: 109608 ***
universal_:
Event - 74883 - rank: universal
arrest/1 - 820 -
subject_:
police/2 - 172748 - rank: universal
action_:
arrest/4 - 823 -
object_:
man/2 - 79018 - rank: individual
time_:
present_ - 248575 -
object_of:
Event - 170891 - rank: individual
Event - 172767 - rank: individual
Event - 172779 - rank: individual
****
event: Police arrests a man.

*** police/2: 172748 ***
generalisation_:
police/2 - 171402 - rank: universal
subject_of:
arrest/1 - 109608 - rank: individual
suspicion/1 - 172745 - rank: individual
****
object: Police.

*** Event: 172767 ***
universal_:
Event - 74883 - rank: universal
subject_:
Event - 79016 - rank: individual
action_:
during/2 - 61250 -
object_:
arrest/1 - 109608 - rank: individual
****
event: A man having age 59 during police arresting him.

Figure 1 - Examples of events and objects used for representing deep semantics

4 Conclusions

The present goal is to provide a requirement analysis that can be easily understood and checked by a human, using graphic displays and NLP query answering. Furthermore, because the internal representation is a well-formalized graph, the analysis results could also be directly fed into the next stage of system development.

The roadmap is as follows: increase the ability of the hand-crafted system; increase the set of gold models analyses (i.e. results of system analysis improved by hand); use this with a suitable fitness function [Kra17] to improve large scale testing of the system; improve the system itself by deploying genetic algorithms based on the existing system, the fitness function and the gold models. All these are being developed using parallel architectures (Erlang and server-less clouds). At the same time, user-friendly interfaces are being developed.

References

[Abb83] Abbot R. Program design by informal English descriptions. *Comm. ACM* 26(11): 882-894, 1983.

[Bäu17] Bäumer F.S., Dollmann M., Geierhos M. Studying software descriptions in SourceForge and app stores for a better understanding of real-life requirements. *WAMA2017*: 19-25, 2017. Doi: 10.1145/3121264.3121269

[Boo94] Booch G.*Object-oriented analysis and design with applications*. Benjamin/Cumming, 1984.

[Bog95] Boguraev,B., Garigliano R., Tait J. Editorial. *Journal of Natural Language Engineering* 1(1):1–7, 1995.

[Bur95] Burg J.F.M., Van de Riet R.P. COLOR-X: Object modeling profits from linguistics. *KB&KS95*: 204-214, 1995.

[Bur97] Burg J.F.M. *Linguistic instruments in requirements engineering*. IOS Press, 1997.

[Fer16] Ferrari A., Spoletini P., Gnesi S. Ambiguity and tacit knowledge in requirements elicitation interviews. *Requirements Engineering* 21(3):333-355, 2016.

[Gén13] Génova G., Fuentes J.M., Llorens J. et al. A framework to measure and improve the quality of textual requirements. *Requirements Engineering* 18(1): 25-41, 2013. Doi 10.1007/s00766-011-0134-z

[Gov09] Governatori G. (Ed.) *Legal knowledge and information systems*. IOS Press, 2009.

[Joh06] Johnson-Laird P. *How we reason*. Oxford University Press, USA, 2006.

[Kiy08] Kiyavitskaya N., Zeni N., Mich L, Berry D.M. Requirements for tools for ambiguity identification and measurement in natural language requirements specifications. *Requirements Engineering* 13(3):207-239, 2008.

[Kiy09] Kiyavitskaya N., Zeni N., Cordy J.R., Mich L., Mylopoulos J. Cerno: Lightweight tool support for semantic annotation of textual documents. *Data Knowledge Engineering* 68(12):1470-1492, 2009.

[Kra17] Kramer O. *Genetic Algorithm Essentials*. Springer, 2017.

[Len90] Lenat D.B., Guha R.V., Pittman K., Pratt D., Shepherd M. Cyc: toward programs with common sense. *Comm. of ACM* 33(8): 30-49, 1990. Doi: 10.1145/79173.79176 http://doi.acm.org/10.1145/79173.79176

[Mic00] Mich L., Garigliano R. Ambiguity measures in requirements engineering. *ICS2000,* Beijing: Publishing House of Electronics Industry, pp. 39-48, 2000.

[Mic02] Mich L., Garigliano R. NL-OOPS: A requirements analysis tool based on natural language processing. *Data Mining III,* Southampton: WIT Press, pp. 321-330, 2002. Doi: 10.2495/DATA020321

[Mic04] Mich L., Franch M., Novi Inverardi P.L. Market research for requirements analysis using linguistic tools. *Requirements Engineering* 9(1):40-56, 2004. Doi: 10.1007/s00766-003-0179-8

[Mic94] Mich L., Garigliano R. A Linguistic approach to the development of object oriented systems using the natural language system LOLITA. *ISOOMS 1994*: 371-386, 1994.

[Mic96] Mich L. NL-OOPS: from natural language to object oriented requirements using the natural language processing system LOLITA. *Natural Language Engineering* 2(2):161-187, 1996.

[Rum91] Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorensen W.E. *Object-oriented modeling and design*. Englewood Cliffs, NJ, Prentice-hall, 1991.

[Rya93] Ryan, K. The role of natural language in requirement engineering. *ISRE:* pp. 240–242, 1993.

[Tjo13] Tjong S.F., Berry D.M. The Design of SREE - A Prototype Potential Ambiguity Finder for Requirements Specifications and Lessons Learned. *REFSQ2013:* pp. 80-95, 2013.

[Zen07] Zeni N., Kiyavitskaya N., Mich L., Mylopoulos J., Cordy J.R. A lightweight approach to semantic annotation of research papers. *NLDB2007*: pp. 61-72, 2007. Doi: 10.1007/978-3-540-73351-5_6

[Zen15] Zeni N., Kiyavitskaya N., Mich L., Cordy J.R., Mylopoulos J. GaiusT: supporting the extraction of rights and obligations for regulatory compliance. *Requirements Engineering* 20(1):1-22, 2015.

[Zen17] Zeni N., Mich L., Mylopoulos J. Annotating legal documents with GaiusT 2.0. *IJMSO:* 12(1):47-58, 2017.