Natural Language Processing of Patents and Technical Documentation

Gaetano Cascini[1], Alessandro Fantechi[2], and Emilio Spinicci[2]

[1] Dipartimento di Meccanica e Tecnologie Industriali
cascini@ing.unifi.it
[2] Dipartimento di Sistemi e Informatica
Università degli Studi di Firenze
Via di S. Marta, 3 - 50139 Firenze, Italy
{fantechi,spinicci}@dsi.unifi.it

Abstract. Natural Language Processing techniques for text-mining and information retrieval are finding application in the analysis of many kinds of documentation, from technical documentation to World Wide Web. Particularly, Functional Analysis techniques are based on the extraction of the interactions between the entities described in the document: these interactions are expressed as Subject-Action-Object (SAO) triples (obtainable using a suitable syntactic parser) which represent a concept in its most synthesizing form. In this work, the techniques developed for a functional analysis of patents and their implementation in the PAT-Analyzer tool are presented. The same technique has been properly tailored and applied to the analysis of software requirements documents. Current work in the direction of the development of a SAO-based Content Analysis of technical documentation is presented.

1 Introduction

Text Mining and Knowledge Management technologies are assuming a key role for many organizations: in order to propose competitive products or services it is necessary to minimize the resources dedicated to the accomplishment of repetitive tasks and to focus on "creative" activities. Moreover, innovation is basically limited by psychological inertia on one hand, and by lacks of knowledge on the other.

Therefore, information retrieval, documents classification, business intelligence, technology forecasting, competitors monitoring etc. are nowadays crucial activities requiring advanced tools capable to face the dramatic paradox that comes out from the availability of huge amount of data from a bewildering variety of sources: an overload of information means no usable knowledge. Such a contradiction between the width of the information source and the low usability of a large amount of documents is typically met in patent search activities: monitoring competitors, checking the novelty of an invention or looking for technical solutions in other fields of application require big efforts even for skilled researchers. Performing a text analysis for constructing design representation has been approached by several authors, as in [8], by means of techniques mainly based on statistics (i.e. counting terms frequencies,

S. Marinai and A. Dengel (Eds.): DAS 2004, LNCS 3163, pp. 508–520, 2004.

identifying specific words etc.). The same approach is followed by tools specifically dedicated to patents analyses as [19, 26], but their main limit is that they cannot distinguish the role of a component in a technical system.

The commercial availability of software capable of analyzing documents with semantic processing algorithms offers a revolutionary way to search, summarize and classify information. Linguistic Analysis tools allow the identification of the key elements of a document, by combining morphological, syntactic and semantic analyses.

Major results can be obtained by analyzing structured documents whose format is strictly related to the content. This characteristics is typical of several forms of technical documentation, and allows "low-cost" Linguistic Analysis techniques and tools to be adopted. Patents are a typical example, with distinguished sections for claims, background of the invention, description of the preferred embodiment etc.

The application of Linguistic Analysis techniques to patent documents, combined to the knowledge about patent format, allows patents analyses and comparisons [5, 6] to be speeded up, providing automatically a functional description of an invention. By means of syntactic parsers we can identify Subjects, Actions and Objects of a sentence (SAO triads). Subjects and Objects may refer to components of the system, Actions may refer to functions performed by and on components. We are interested to select those SAOs in which the Subject is the Tool performing a function on the Artifact referred by the Object, and the action is the Field that links the Tool and the Artifact (TFA triads). The successful application of Natural Language techniques to patent documents and the development of a specific tool (PAT-Analyzer), described in section 2, has prompted the authors to investigate the portability of the approach to other forms of technical documentation. Indeed, in the field of Software Requirement Engineering, several attempts have been made in the same direction, and section 3 surveys some of them. In this case, the aim of the analysis of technical documentation is the discovery of possible ambiguities or incompleteness on one hand, and the detailed comprehension of the requirements in the direction of a possible formalization on the other hand. In section 4 the paper presents the guidelines along which the approach and tool developed for the analysis of patents have been adopted and modified for the analysis of software requirements documents, as well as a novel way of performing Content Analysis both on Patents and Software Requirement documents.

2 Patents Functional Analysis

Functional Analysis is a powerful tool for conceptual design both for problem identification and innovative solutions generation: the functional description of a product is a description at an abstract level, so that different design solutions can be explored by developing functional variants. Moreover, functional analysis helps the designer in following a systematic approach also in the study of complex systems, by breaking up functions into simpler subfunctions and subdividing the problem into more manageable parts. Finally, functional analysis can play an important role also in activities of patent breaking [12]. Inversely, when writing a new patent, a text-based functional analysis is an effective test of the suitability of the work done.

Further advantages can be obtained by performing functional analyses with a TRIZ-based way of thinking, i.e. representing not only useful functional relationships between system components, but also harmful, ineffective, excessive and missing functional relationships [16].

Functional modeling is used also at a detailed design stage: following the Suh approach [27] the function is the desired output and the design is decomposed into functional requirements which are mapped directly with the design parameters at any abstraction level.

Several works have proposed comprehensive representations of functions which represent the different aspects of the designers' intention, that is a crucial issue for developing computer aided conceptual design systems; the aim of these works is defining effective ways to represent also the relationships among the functions, i.e. decomposed-into, conditioned-by, enhanced-by and described-as relations [22].

In this context the authors are developing tools and methods to help the designer in performing functional analyses making use of several kinds of inputs. As a first attempt, the procedure has been optimized to define the functional diagram of a US Patent [23, 14], since those documents must follow a comprehensive set of rules [20]; hence it is possible to tune a semantic processing algorithm such that its output can be suitably post-processed in order to build the functional scheme of the analyzed patent. This procedure has been implemented in a software tool, PAT-Analyzer [6], capable to identify the components of the patented system, perform a hierarchical classification of those components subdividing them in different abstraction levels, draw a functional diagram of the complete system and of the detailed subassemblies, therefore providing a graphical representation of the invention described in a patent. By means of several types of score ranks it is possible to highlight the core of the invention, the most peculiar components and performed functions.

2.1 PAT-Analyzer Methodology

The data flow is represented in Figure 1: the text analysis consists of three steps aimed at (i) identifying the components of the invention; (ii) classifying the identified components in terms of detail/abstraction level; (iii) identifying positional and functional interactions between the components both internal and external to the system. Several types of analysis can be performed by means of the post-processing module, in order to focus on the invention peculiarities. The components identification task is performed taking into account that all the components must be numbered univocally to be identified in the illustrations [22]. A lemmatizer and a set of filters and synonyms can be adopted in order to improve the quality of the results.

Therefore, a list of reference denominations and alternative denominations is extracted for each component. The following analysis is dedicated to the search of descriptive locutions (i.e. sentences containing verbs like "to form", "to constitute" etc.) and specification's expressions (like "the gripper of the pivot arm") in order to identify subsystem/supersystem relationships, hence defining a hierarchy of detail/abstraction levels (Figure 2, left).

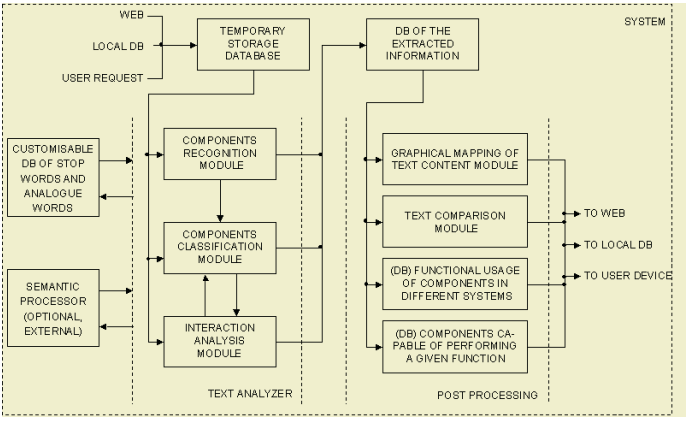

Fig. 1. PAT-Analyzer data flow.

Finally, positional and functional interactions between the identified components are determined by filtering, from the list of subject-action-object (SAOs) provided by a syntactic parser, the triads containing irrelevant verbs (Figure 2, right). Further details about the algorithm can be found in [5].

The whole set of extracted information can be synthetically represented by means of a diagram according to the following set of rules:

```
induction-heating bender 1
|--- conveyor means 10
|    o--- conveyor 11
|         o--- rolls 12
|--- pivot arm 20
|    |--- gripper 21
|    |    |--- stationary table 22
|    |    |--- turntable 25
|    |    o--- manually-operated jack 27
|    |--- stud pin 23
|    |--- chuck device 24
|    |    o--- turntable 25
|    o--- catch piece 26b
|--- arm support 30
|    |--- screw drive 31
|    |--- guide rails 32
|    |--- carrier 33
|    |    o--- square holder 37
|    |              o--- pivot 36
|    o--- speed-reduction unit 35
|--- heating-and-cooling mechanism 40
|    |--- coil unit 41
|    |    o--- coil sub-unit 42
|    |         |--- cooling compartments 46
|    |         o--- ejection nozzles 48
|    |--- coolant pipe 47
|    |--- heating-and-cooling mechanism 50
|    |    |--- square pipe 51
|    |    o--- coil unit 53
|    o--- heating-and-cooling mechanism 60
|         |--- triangular pipe 61
|         o--- coil unit 63
|--- motor 34
|--- transformer 43
o--- support arm 44
```

Tool	Field	Artifact
cooling compartments 46	effect	different hardenings
elongate slot	cool	cooling compartments 46
conveyor means 10	feed	workpiece
gripper 21	grip	workpiece
…	…	…

Fig. 2. Preliminary results obtained by the analysis of the patent US 6,097,012 [23]: hierarchical classification of the components (left) and list of functional interactions (right).

1) each identified component of the system is represented by a rectangle labeled with its reference number and the representative name defined in the Components Recognition phase; each identified component or subject external to the system is represented by a gray rectangle labeled with a representative name;
2) the detail level hierarchy is represented nesting the components at a deeper detail level inside the components at a more abstract level;
3) the functional interactions between the identified components are represented with straight arrows pointing from the Tool to the Artifact, labeled with the Field;
4) the positional interactions between the identified components are represented with dashed arrows pointing from the Tool to the Artifact, labeled with the Field.

The post-processing module allows the identification of the most relevant concepts contained in the patent text by means of several metrics (examples of application will be provided in the next section of the paper):

1) Detail Level Chart: on the basis of the components hierarchical classification performed in the text analysis phase, it is possible to assign a Detail Level (DL) to each TFA triad and/or to each paragraph: a DL is assigned to each component so that the maximum abstraction level is represented by a DL=0 and the DL of each subsystem is one level greater than the DL of the corresponding supersystem. The detail level of a basic sentence (TFA) is estimated as the average DL of Tool and Artifact; the detail level of a paragraph is the average DL of all the TFAs belonging to that paragraph. By analyzing the DL run along the description it is easy to identify the most relevant paragraphs and concepts of the inventions (it is worth to notice that if the patent description is focused on specific details of the proposed system, it means that such a part of the description is strictly related to the peculiarities of the invention).
2) Components Recurrence Analysis: by counting the citations of each component of the proposed invention (of course taking into account of all the alternative denominations) it is possible to determine a relevance score of the components themselves; in order to improve the quality of such an estimation, different weights are assigned to the citations found in the title, abstract, independent claims, dependent claims, summary, description of the preferred embodiment.
3) TFA Recurrence Analysis: by means of the same technique adopted for the Component Recurrence Analysis, it is useful to assign a score to the functional and positional triads TFA. Also partial pairs of the triads (TF, FA, and TA) are counted by means of the same set of weights. Therefore a rank of the most relevant sub-functions performed by the invention is determined, again in order to focus the attention of the user to the patent peculiarities. The partial pairs score allows to extract complementary information: for example, if the FA score is greater than the corresponding TFA, it could mean that the same component receives the same function by several tools, i.e. a “combination” has been used to reinforce the effect of the action. Besides, a TF score greater than the corresponding TFA, could mean that a parts count reduction is attempted by integrating functionalities in the Tool, in order to reduce costs.

2.2 An Example of Patent Analysis

A typical problem encountered while monitoring the intellectual property of competitors is the increasing number of patents being presented with either very simple or very non-informative titles. The strategy behind such titles is that they are unlikely to attract interest. Moreover, very often even the drawings are not so explicative.

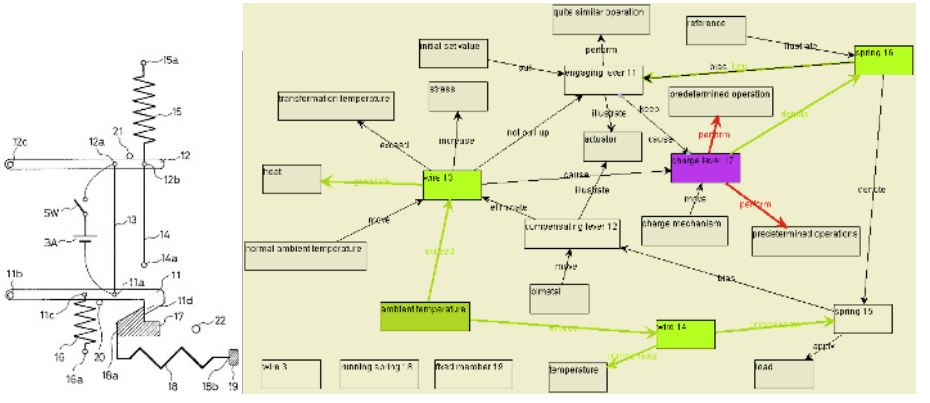

Fig. 3. US Patent 6,459,855 "Actuator" [14]: first drawing of the patent (left); functional diagram and TF recurrence top scores (right).

A tool capable of identifying the core of the invention automatically is therefore very useful to check the relevance of a patent with minimal efforts.

As an example let's take into account the US Patent 6,459,855 "Actuator" [14], assigned to Minolta: the patent is 35 pages long and describes several embodiment variants; therefore understanding the real purpose of such an invention and its relevance requires a long and careful reading. The drawings are not as meaningful as shown in Figure 3, left: what is the function of the wires 13 and 14?

The proposed methodology speeds up the comprehension of the system without the need of reading the patent text, but just combining the functional diagram with the drawings of the patent. In this case the list of components identified by PAT-Analyzer has been reduced by deleting the components not present on the main drawing.

The resulting diagram is shown in Figure 3, right, where the most recurrent Tool-Field pairs have been highlighted. While in such a case the detail level run analysis would not provide any useful information, since the components hierarchy is absolutely flat, looking for the most cited interactions in terms of subject-function points directly to the invention peculiarity: the role of the wires 13 and 14, absolutely not understandable by means of the drawing alone, is explained by the TFAs "ambient temperature" – "exceeds transformation temperature of" –"wire 13/14". These TFAs clearly suggest that shape memory alloys have been used in order to "not pull up" – "engaging lever 11", to "compensate" – "spring 15", to "cause" – "charge lever 17" – "perform" – "predetermined operations".

It is worth to note that applying statistical analyses to pre-processed data (in this case the identified components and their interactions) is much more effective than simply counting terms occurrences as usually performed by standard text mining tools.

3 NLP Techniques for Software Requirements Documents

Requirement Engineering is the branch of the software engineering which deals with a proper definition of the features a software should have to satisfy the needs of the target users. Software Requirements represent the agreement, often with a legal status, between the software developers and their customers: for this reason, on one hand they should be comprehensible by non-technical people, on the other hand they should precisely define the functionalities of the software to be developed, and they should be able to drive the later stages of the software development. They are specified in Natural Language, which is comprehensible but may introduce ambiguity.

A well structured requirement engineering process can be typically divided in three steps: first, general information about the problem to be solved are collected from the customers of the software product, producing a description in natural language; starting from the previous document, the requirement specification is made, which is a detailed description of the product to be implemented; finally, a software specification is produced to abstractly describe the software system to be implemented. Through these steps a rich, informal and potentially ambiguous natural language description should be converted to a formal and essential one. Actually, in many software houses requirement engineering results in only one document in which the system to be implemented is described, often using natural language only, with the risk of obtaining an unsatisfactory product.

Revealing the sentences affected by ambiguity in a software requirement description can be achieved by analyzing the sentence using NLP techniques from a lexical or syntactical point of view: for this reason, it is proper to talk about, for example, lexical non-ambiguity or syntactic non-ambiguity rather than non-ambiguity in general. For instance, a sentence may be syntactically non-ambiguous but it may be lexically ambiguous because it contains wordings that have not a unique meaning.

Lexical evaluation uses lexical parsers to detect and possibly correct terms or wordings that are ambiguous (i.e. that may have multiple meanings according to the context): the tools *QuARS* [9] and *ARM* [30] belong to this category. These tools can be seen as an aid for "writing the requirements right," not "writing the right requirements": they provide structural and quality indicators on the basis of a suitable model, defined considering the existing literature and experiences in the field of requirement engineering and software process assessment: the SPICE (ISO/IEC 15504) model [13], adopted by *QuARS*, provides for example lexical defect indicators for the *Testability* of a requirement description in terms of *vagueness* (that is, a sentence contains words having a non uniquely quantifiable meaning), *subjectivity* (if a sentence refers to personal opinions) *optionality* (a sentence containing one or more optional terms), and so on. For example, the sentence "The C code shall be clearly commented" is

pointed as vague because of the vagueness indicator “clearly”. The tool points out these defects without forcing any corrective actions, leaving the user free to decide whether modifying the document or not. The use of targeted dictionaries allows the sentences to be analyzed taking into account the particular application domain.

Syntactical evaluation exploits syntactical analyzers to detect sentences having different interpretations, as in the case of tools *LOLITA* [18, 17] and *Circe-Cico* [1, 2].

LOLITA is a general purpose NLP tool that is able to identify the morphological, grammatical, semantic and pragmatic relations of a sentence, by representing them with a semantic net, in a similar way as a concept graph, which is used as a knowledge base of the system: the nodes composing the net are hierarchically connected. The goal is to reduce the ambiguity of a sentence using statistical techniques, in order to identify the most likely interpretation of a requirement. The number of extracted parsing trees is used to compute ambiguity measures, which allow also for weights that are assigned accordingly to the construction criteria of the trees themselves.

The *Circe-Cico* environment provides a support for identifying, selecting and validating NL requirements, with the integration of information from the application domain. The *Circe* architecture is constituted by five components: a *semantic database* containing small units called *atomic requirements;* the *modeler* components generate a high-level model starting from the atomic requirements, that can also be decomposed in order to store each part in suitable *syntactic archives*, used together with the modelers to store lexical, syntactic and pragmatic contents; the *projector* components build the actual representation and parse the user input; there is a projector for each kind of model, which is represented using a suitable intermediate language. A *translator* component produces a representation of the model accordingly with the language for the chosen representation (graphs, tables, and so on).The *Cico* component is a projector which produces parsing trees of the input NL documents: based on a fuzzy algorithm, it also uses backtracking and heuristics to optimize the parsing tree generation, and only lexical correctness of the sentence is required. In its latest version [3, 4], the *Circe-Cico* tool allows the formalization of NL requirements in Abstract State Machines (ASM) or UML diagrams. *Circe-Cico*, as well as *LOLITA*, requires to the user a considerable familiarity.

Some recent studies addressing ambiguity and interpretation problems have dealt with a relation-based approach applied to a requirement document structured accordingly to the *Use Case* formalism [7], which provides for the definition of entities (the *Actors*) interacting each other and with the described system. Particularly, the studies presented in [10, 11] investigate methods to provide support for *Consistency* and *Completeness* checking, that is to detect the presence in the NL requirement document of semantic contradictions and structural incongruities, by means of extracting the relations between actors and the system.

The experience of patents functional analysis suggests the extraction of SAOs as a basis for an alternative method to verify consistency and completeness of requirement definition; a twin tool (J-RAn, Java Requirement Analyzer) of the PAT-Analyzer is currently being implemented with the introduction of a novel SAO-based Content Analysis, both described in the next section.

4 SAO-Based Content Analysis

Content Analysis has been long recognized as a mean for text analysis and information retrieval; a milestone is the work of Siemens [24], where the basic techniques for content analysis are explained: the use of dictionaries and glossaries of keywords allows to perform a statistics of their occurrences in a text, and to detect if the content of a document deals with a topic of interest; the located document sections can be tagged to improve searching operations. This kind of text mining technique has found a wide application in the Communication Science [15, 28]; its basic principles can also be found in the IBM's WebFountain [29], to cite the most recent text application to the analysis of web documents. Recurrence analysis can be evolved in a novel SAO-based Content Analysis technique for the investigation of the parts of a document (either a patent or a requirement one) dealing with specific topics, as described in the following subsections.

The SAO-based analysis allows searching for "key-concepts" instead of keywords, therefore providing a higher efficiency to the document analysis tools.
The proposed technique consists in using suitable dictionaries of verbs and terms related to a topic of interest, and to assign a score to each SAO if its subject, verb and object belong to the adopted dictionaries. Giving different weights to subject, verb and object, it is possible to put in evidence functional relationships or terminology accordingly to the purpose of the analysis. The previous weights can vary depending on the type of document under analysis.

The average value of SAOs score in a sentence (or in a whole paragraph) gives a measurement of how this sentence (or paragraph) deals with the topic of interest. This information can be traced in a suitable (and possibly normalized) chart showing the score vs. sentence/paragraph number, to obtain an overview of the entire document about the parts treating the topic of interest, and in which measure.

4.1 Requirement Analyzer Tool

The realization of the J-RAn tool has started by the observation that the extraction of SAO relations can constitute a basis of transferable techniques for the analysis of both patents and software requirements. J-RAn is able to extract the descriptive paragraphs related to each software requirement, by searching each requirement tag matching a user-defined format, which can vary depending on the document model. The extraction of paragraphs and corresponding sentences is implemented using Phrasys NLP components [21]. The SAO extraction is then performed by a syntactic parser (in the first version, LinkGrammar [25]).

J-RAn inherits from PAT-Analyzer the recurrence analysis technique as a mean to detect possible inconsistencies, by performing the SAO-based Content Analysis for example to search whether requirements corresponding to a given functionality are expressed only in the desired sections of the document: if other paragraphs are dealing with the functionality under discussion, the document probably contains a redundancy, that is source of possible inconsistencies.

Current experiments on different software requirement documents use a score assignment in which the functional relationship is favoured (weights: Subject:3, Verb:5, Object:2). An example of resulting chart is reported in Figure 4.

An accurate experimentation of SAO weights will result in the most suitable values also for other kinds of technical documentation (such as handbooks).

4.2 Patent Analysis

The content analysis approach can be followed also with the purpose of enriching information extracted by patents. Nevertheless, in such a case the use of standard dictionaries and glossaries is unadvised, since it precludes the retrieval of breakthrough solutions (that typically introduce new technologies in a given field of application). Moreover, it is worth to remember that the relevant concepts of the invention can be extracted by means of the techniques described in section 2. Therefore, the proposed methodology consists in performing occurrence searches in the patent text, by looking for the top score components and/or TFA triads and TF/FA pairs.

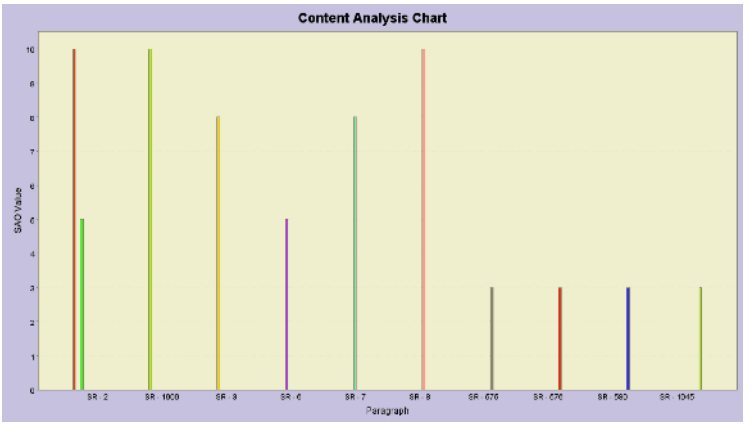

Fig. 4. Content Analysis chart related to a Software Requirement document.

It has been observed that such an approach is much more effective than traditional terminology analysis based on stemming and statistical analysis of the extracted lemmas. In facts, a word or even a multi-word can be mentioned several times in a patent, nevertheless not representing the peculiarity of the invention. (It should be mentioned that also the "to be" auxiliary verb has of course a big number of occurrences, but "general" terms are usually filtered by means of a set of stop words). In the case of the actuator claimed in [14], the most cited multiword is "shape memory alloy" with 250 occurrences. A traditional text processing technique highlights that the invention is related to the use of shape memory alloy properties, but doesn't provide any information about how and why such technology has been adopted, since it is mentioned all over the document (Figure 5).

Besides, the PAT-Analyzer recurrence analysis identifies the “wire 13” as the most important component of the invention together with the “engaging lever 11” (94 and 91 occurrences respectively, apart from the weight of the part of the document where they are extracted from). The paragraphs where “wire 13” is mentioned are less than those containing the multiword “shape memory alloy” (Figure 5), thus focusing the attention of a patent analyst on a more meaningful portion of the invention description. Moreover, PAT-Analyzer identifies two SAOs (Figure 4) as the most relevant: “ambient temperature – exceed – transformation temperature of wire 13” and “ambient temperature – exceed – transformation temperature of wire 14”: by executing a SAO based content analysis searching these two triads it is possible to point directly to the paragraphs where the invention peculiarity is described.

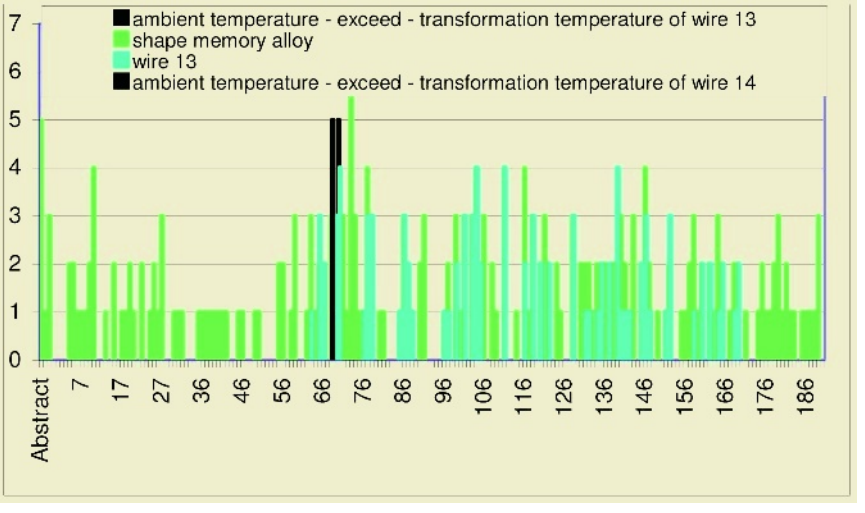

Fig. 5. Example of Content Analysis chart related to patent [14].

5 Conclusions

In this paper it has been shown how the use of available NL parsers, which are anyway not particularly sophisticated, can support functional analysis techniques on technical documentation, such as recurrence and content analysis. The experiments carried out on Patents and Requirement documents show that these techniques are able to effectively support the human comprehension of these classes of technical documentation.

Regarding in particular requirement documents, current work deals with the implementation in J-RAn of a feature for expressing the extracted functional relationships in the XMI format, in order to export them to commercial UML tools (Rational Rose, Visual-Paradigm), capable of visualizing the relationships as a UML Class Diagram.

As a future work, SAO-based Content Analysis will be experimented on other kinds of documentation, either technical (i.e. handbooks), or not (i.e. web and public administration documents), with a suitable tuning of weights depending on the application.

References

1. V. Ambriola and V. Gervasi. *An environment for cooperative construction of natural-language requirement bases*. In Proc. of the 8th Conference on Software Engineering Environments, pages 124-130. IEEE Computer Society Press, Mar. 1997.
2. V. Ambriola, V. Gervasi, *Experiences with Domain-Based Parsing of Natural Language Requirements*, Proc. 4 th International Conference NLDB '99, Klagenfurt, Austria, 1999.
3. V. Ambriola, V. Gervasi, *On the parallel refinement of NL requirements and UML diagrams.* In Proc. of the ETAPS 2001 Workshop on Trasformations in UML, Genova, Italy, Apr. 2001.
4. V. Ambriola, V. Gervasi, *Synthesizing ASMs from natural language requirements*. In Proc. of the 8th EUROCAST Workshop on Abstract State Machines, Feb. 2001.
5. G. Cascini, V. Abate, D. Lucchesi, P. Rissone: *System and Method for performing functional analyses making use of a plurality of inputs*. Patent Application 02425149.8, European Patent Office, 14.3.2002, International Publication Number WO 03/077154 A2 (Sept. 18, 2003).
6. G. Cascini, P. Rissone: *PAT-Analyzer: a tool to speed-up patent analyses with a TRIZ perspective*. Proc. of the ETRIA World Conference: TRIZ Future 2003, Aachen, Germany, Nov. 12-14, 2003.
7. A. Cockburn, *Writing effective Use Cases*. Addison-Wesley, 2000.
8. Dong, A. M. Agogino, *Text Analysis for Constructing Design Representation*. Journal of Artificial Intelligence in Engineering, Vol. 11 (2), 1997.
9. F. Fabbrini, M. Fusani, S. Gnesi, G. Lami, *An Automatic Quality Evaluation for Natural Language Requirements.* REFSQ'01 International Workshop, Interlaken, Switzerland, Jun. 2001.
10. A.Fantechi, S.Gnesi, G.Lami, A.Maccari, *Application of Linguistic Techniques for Use Case Analysis*. Requirements Engineering Journal, Vol. 8, Issue 3, pp. 161-170, Springer-Verlag, Aug. 2003.
11. A.Fantechi, S.Gnesi, G.Lami, *A Relation-based Approach to Use Case Analysis*. Proceedings of the 9th International Workshop on Requirements Engineering: Foundation for Software Quality - REFSQ03, Velden, Austria, Jun. 16-17 2003.
12. S. Ikovenko, *Patent Breaking*. Invention Machine 4th Annual European User Group Meeting - Bergamo, Italy, 24-26/9/2000.
13. ISO/IEC TR 15504 (Parts 1-9), 1998.
14. A. Kosaka et al., *US Pat. 6,459,855 – Actuator*, http://www.uspto.gov/patft/index.html, Oct. 1, 2002.
15. K. Krippendorff, *Content Analysis : An Introduction to Its Methodology*, 2nd Edition. Sage Publications, Dec. 2003.
16. D. Mann, *Hands On Systematic Innovation*. CREAX, 2002.
17. L. Mich, Ambiguity *identification and resolution in software development: a linguistic approach to improve the quality of systems*. In Proc. WESS'01 International Workshop, Firenze, Nov. 2001.

18. L. Mich, R. Garigliano, *Ambiguity Measures in Requirement Engineering*. Int. Conf. On Software Theory and Practice - ICS 2000, Beijing, China, Aug. 2000.
19. F. Neri, R. Raffaelli, *A new way of exploring patent databases*. http://www.synthema.it, 2003.
20. *Patent Rules: Title 37 - Code of Federal Regulations - Patents, Trademarks, and Copyrights*, http://www.uspto.gov/patft/index.html, last update Dec. 18, 2000.
21. *Phrasys Natural Language Processing Software on-line*. See: http://www.phrasys.com.
22. Y. Shimomura, M. Yoshioka, H. Takeda, Y. Umeda, T. Tomiyama, *Representation of Design Object Based on the Functional Evolution Process Mode*. Journal of Mechanical Design (ASME), Vol. 120, Jun. 1998.
23. K. Shiozuka, *US Pat. 6,097,012 - Induction-heating bender*. http://www.uspto.gov/patft/index.html, Aug. 1, 2000.
24. R. Siemens, *Practical Content Analysis Techniques for Text-Retrieval in Large, Un-tagged Text-bases*. Presented at the ACM SIGDOC' 93 Conference, University of Waterloo, Oct. 1993.
25. D. D. K. Sleator, D. Temperley *Parsing English with a Link Grammar*. Third International Workshop on Parsing Technologies, Aug. 1993.
26. A. Spinakis, G. Panagopoulou, A. Chatzimakri, *STING: A Text Mining Tool supporting Business Intelligence*. NEMIS Annual Conference, University of Rome "La Sapienza", 23.1.2004.
27. N. P. Suh, *The Principles of Design*, Oxford Press, 1990.
28. R. P. Weber, *Basic Content Analysis*. Sage Publications, Aug. 1990.
29. *WebFountain*, http://www.almaden.ibm.com/webfountain.
30. W. M. Wilson, L. H. Rosenberg, L. E. Hyatt, *Automated Analysis of Requirement Specifications*. ICSE'97, Boston, MA, May 1997.