NLP-based Patent Information Retrieval

Olga Babina
South Ural State University
olga_babina@mail.ru

Abstract

The paper presents an approach to information retrieval for a specific domain of patent claims. The approach rests on utilizing predicate-argument structure representation of full-text documents for organizing a fine-tuned retrieval of semantic information. Indexing and search rely on a versatile data from a thoroughly elaborated linguistic knowledge base. The procedure of automatic indexing resulting in forming the predicate-argument structure of a patent claim is presented. The predicate-argument structure is conceived as a structural unit of the indexed document. The search strategy based on mapping predicate-arguments structures of the documents and query representations is described.

1. Introduction

Contemporary information retrieval (IR) systems have to deal with information presented as natural language texts. It has become evident that simple string-matching is insufficient for intellectual retrieval of documents *semantically* relevant to the user's requirements. Natural language is an inevitable medium (the most powerful of possible) to present information in a document database and to make explicit a user's information need. To handle full-text documents the knowledge base should be thoroughly elaborated. The approach exploited for solving the information retrieval task is based on the idea that NLP-techniques can benefit for the performance of IR systems.

The attempts to improve retrieval performance through more sophisticated linguistic processing have been taken for the recent years [see Strzałkowski, 1994; Smeaton, 1999; Voorhees, 1999; Mihalcea and Moldovan, 2000; etc]. Notwithstanding the fact that comprehensive approaches are desirable, it is widely accepted that linguistically-oriented systems work better if they are designed for a particular domain. This arises from the intuitively obvious fact that specific domains are restricted in terms of semantics, grammar and lexical ambiguity. In general language a researcher has to handle a complicated network of relations (not always explicit) between its elements and has to tackle a huge number of issues connected with numerous cases of ambiguity. The language model for such systems should be extra carefully elaborate which is consuming in terms of time and costs and there is no guarantee that the researcher won't miss anything and that the model will cover all the phenomena characterizing general language. This inevitably leads to poor effectiveness of those considerably error-prone systems.

A kind of solution for this crucial problem is restricting to a certain domain. Constraints being laid on the language processed make it simpler and easier to process. This also refers to IR systems enhanced by modules based on natural language processing (NLP) techniques. In this research the domain under consideration is patents, namely patent claims for method.

A patent claim is a certain part of a patent document having the economical, technical and legal power. It comprises the description of the most essential peculiarities of an invention. A patent claim is a unique linguistic object having specifically organized structure and a rather complicated syntax. An example of a US-patent claim is given in Figure 1.

A method for using a cDNA to detect the differential expression of a nucleic acid in a sample comprising:

a) hybridizing the probe of claim 4 to the nucleic acids, thereby forming hybridization complexes; and

b) comparing hybridization complex formation with a standard, wherein the comparison indicates the differential expression of the cDNA in the sample.

Figure 1. *An example of a patent claim from the patent US 6,485,910.*

As a patent claim is the most informative part of a patent document it seems sensible, when performing patent expertise in patent offices, to exploit patent claims descriptions at the stage of searching patents potentially infringed by a submitted invention description. For the purpose a fine-tuned parser at the stage of indexing and a set of metrics constituting our search strategy are applied, all resting on a thoroughly elaborated linguistic knowledge base.

2. Linguistic Knowledge Base

The knowledge base includes lexical resources representing linguistic information about the lexical units. The premise is that a most considerable part of linguistics description should be presented in the lexical knowledge base, the latter thus playing a crucial role in NLP-based IR later.

Linguistic knowledge base comprises the following:

1) **Domain-Specific Lexicon**: contains a versatile data about the lexical units of the domain investigated. Linguistic approach exploited for patent claims language modeling is based on dependency grammar [Tesnière, 1959]. Special attention is paid to predicates, which are conceived as semantic centres of sentences. *Predicates* are understood as the elements of proposition which denote the situation having a certain number of obligatory participants (*arguments*) playing certain roles.

There are two levels of representation for lexical units in the lexicon: *lexico-morphological* and *syntax-semantic*.

Lexico-morphological level of a word representation includes its lemma, part of speech and explicitly specified morphological paradigm of the word. For the purpose of claim analysis active and passive forms of a verb are considered as different predicates, e.g. the predicates *associated* and *associating* are described as two different entries in the lexicon. The inventory of morphological forms to be specified is determined by a grammar category to which the word belongs to, and is restricted to the forms statistically most probable for the appropriate part of speech within the domain. Thus, the lexico-morphological level of active predicates is presented in the lexicon by the forms, corresponding to Participle I Indefinite Active, the Present Indefinite Active Tense, the Present Indefinite Active Tense (3d person singular), the Infinitive Indefinite Active, and the Gerund Indefinite Active.

Syntax-semantic level of representation is rather comprehensive for predicates, though limited for other lexical units by only marking their semantic class. Predicates have *semantic* vs. *surface-syntactic* descriptions at the level. Semantic level is presented by the semantic class of the predicate and its case-roles, constituting the subcategorization frame of the predicate. Subcategorization frames are mostly identical for the representatives of the same semantic class, though variants are possible.

On the surface the case-roles of different predicates can be realized individually. On the one hand, argument fillers vary for different predicates by their syntactic structure. A filler can be a prepositional phrase (PP) at one predicate, and the same argument can be realized as a subordinate clause at another one; or the same case-role of different predicates can be filled by PP, but introduced by means of different prepositions, etc. On the other hand, the linear order of case-roles realization in the text differs even for the same predicate.

Syntactical peculiarities of argument fillers are presented in the lexicon as selectional restrictions (in terms of syntactical phrases) assigned to every case-role within the lexicon entry. Analyzing the corpus it is possible to determine the *linear patterns* [Sheremetyeva, 1999], defining the order of realizing case-roles of a predicate in patent claims. The number of linear patterns for the predicate is rather restricted.

Thus a lexicon entry, for instance of the predicate *comparing*, can be presented as follows:

comparing
[<*lexico-morphological level*>
[LEMMA: comparing]
[POS: predicate active]
[MORPH:
[PI: comparing]
[PRES-IND-ACT: compare]
[PRES-IND-ACT(3SING): compares]
[INF: compare]
[GER: comparing]]
<*syntax-semantic level*>
[SEM-CL: comparison]
[CASE-ROLEs:
[OBJ1: {N NP "NP and NP"}]
[OBJ2: {"with N" "with NP" "to N" "to NP"}]
[MAN: {Adv}]
[PURP: {InfP "for GerP" "so that S"}]]
LINEAR-PATTERNS:
X-OBJ1-OBJ2
X-OBJ1-OBJ2-PURP
MAN-X-OBJ1-OBJ2
X-OBJ1{"NP and NP"}

], where OBJ1, OBJ2, MAN, PURP correspond to the case-roles <object1>, <object2>, <manner> and <purpose> respectively, X in linear patterns denotes the position of the predicate, syntactical selectional restrictions are given in curled brackets.

The lexicon contributes much to the process of NLP-based indexing of patent documents.

2) **Domain-Specific Ontology**: Ontologies are often difined as "an explicit specification of conceptualizations". Existing onologies [see Miller et al., 1990; Fellbaum, 1998; Mahesh and Nirenburg, 1995; Andreasen et al., 2000; etc.] comply with this principle of organization. A domain-specific ontology is ontology representing a structural model of a given domain.

The structural unit of ontology is a concept. In a domain ontology concepts are characteristics of the domain investigated. The concepts are related, at that rather elaborate classifications of relations are suggested. The basic relation utilized in this model is IS-A relation, by means of which the concepts are organized into a subsumption hierarchy.

The domain ontology utilized has a set of beginners corresponding to semantic classes introduced in the domain-specific lexicon, each forming a hierarchy. Each term

of the lexicon is mapped onto one concept in the ontology. Several (synonymic) terms may by mapped on the same concept. Some of the nodes are virtual, and have no terms references.

A model of the ontology oriented for domain 'Pharmacology' may be graphically presented as it is shown in Figure 2[1]. The titles of the nodes are arbitrary attached labels usually chosen as the most frequent among the terms representing the concept in texts.

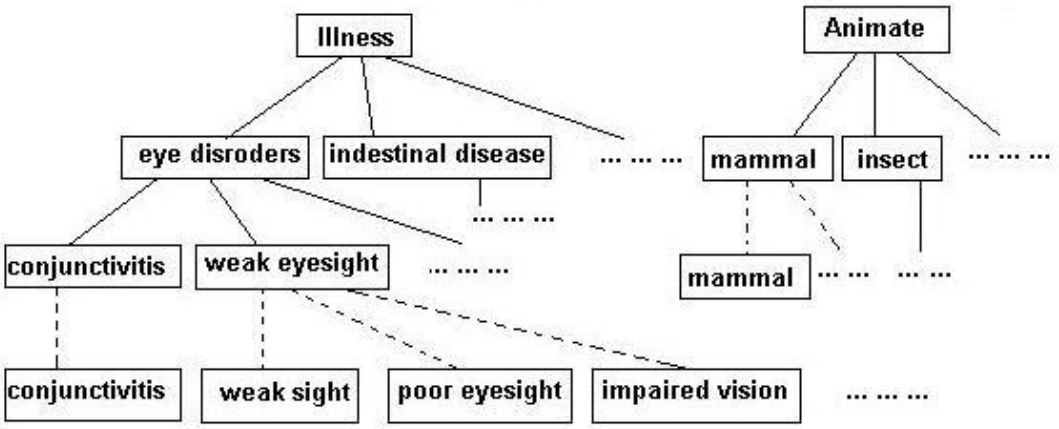


***Figure 2**. An example of an ontology for domain 'Pharmacology'*

For further formal descriptions individual hierarchies are merged into a single hierarchy with a unique virtual identifier TOP, such as introduced in the lexicon semantic classes are its immediate descendants.

The ontology is exploited as the basis of the matching function at the stage of query evaluation.

3. Indexing and Retrieval

3.1. NLP-based Indexing

Indexing is performed by using a full-grammar parser based on the claim analysis model suggested in [Sheremetyeva, 2003] and extended by modules for solving the task of IR. The output of the parser is a representation which reflects predicate-argument structure of text fragments.

Though there is a number of studies devoted to obtaining information about the predicate-argument structure of the sentence from the text [e.g., Surdeanu et al., 2003], the task of doing it automatically is not trivial. The idea of our approach is as follows: knowing what should a filler for a case-role be in terms of syntactic categories and what order the case-roles of a predicate are realized in the text, it is possible, on distinguishing the predicate itself, to find the correspondence between certain parts of the text and the case-roles of the predicate. Thus, when processing the text automatically it is semantic relations that are in focus. Basing on subcategorization frame of the predicate and linear patterns of realizing the case-roles taken from the lexicon one can single out the predicate-argument structure in the text automatically.

[1] Solid lines denote IS-A relations between concepts; dotted lines correspond to the relations between a concept and the terms from the lexicon mapped on the concept (in the computer implementation such relations are not explicitly presented in the ontology, but realized as references to lexicon entries).

Indexing which is intrinsically a parsing procedure consists of the following consecutively performed stages:

1) *Tokenization*: basing on the tabulation, punctuation, and also a frame representing the claim structure the text is divided into parts with the help of "boundary" tags.

2) *Supertagging*: a set of possible supertags taken from the lexicon is assigned to every word. A supertag encodes the semantic class, part of speech and morphological form of the lexical unit. Then, basing on an ordered list of productive rules supertags are disambiguated. All supertags but one are ruled out for the word on the basis of selectional restrictions imposed by the rules. The rules discard wrongly attached tags such that every word is labelled by a single supertag, at that left and right context of two neighbouring words (or supertags) is taken into account.

3) *Bottom-up heuristic parsing*: performs a recursive matching of supertag strings against the patterns of the grammatical component of the stage. Noun phrases (NP), Complex NPs, Prepositional phrases (PP), Adverbial phrases (Adv), Gerundial phrases (GerP), Infinitive phrases (InfP) are successively determined and labelled by an appropriate "phrase" tag as a result.

4) *Co-referencing NPs*: is fulfilled as a result of double passing in the opposite directions along the text lines. Left-right passage is aimed at searching for NPs referring to an antecedent (formally, attributed by the definite article). As soon as a candidate is found left-right passage is temporarily broken and right-left passage is initiated for searching the antecedent. The antecedent NP is retrieved by applying heuristic rules matching the NP in question against potential antecedents. The antecedent having been found, the two NPs are marked as co-referential and left-right passage is resumed until the end of the text is reached.

5) *Detecting semantic dependencies*: includes identifying a predicate in the text and assigning case-roles. While retrieving dependencies the rules for the stage take into consideration the context of 2 phrases left and right from the phrase under consideration.

6) *Predicate-argument structure transformations*: the previously performed stages imply only two-level representation of a predicate-argument structure, i.e. the structure has a predicate centre (first level) and a number of arguments immediately dependent on it (second level). Formally it complies with first-order predicate logic.

Though the text of a claim has a nested structure where one predicate with its arguments may be used as an argument for another predicate (in the exemplified claim (Figure 1) *detect* is embedded into the structure of the predicate *using*; *hybridizing, comparing* are embedded into the structure of *comprising*, etc.). Such syntactic structure is a characteristic of patent claims, and such structures may be even more complicated. In case a predicate has an argument with "situational" interpretation which is presented by means of another predicate construction such argument is transformed by recursively applying the rules of the previous stage to it resulting in a nested predicate-argument structure representation. For example, the predicate construction for *using* having two arguments ('cDNA' and 'to detect the differential expression of a nucleic acid in a sample') from the exemplified claim will be split into the following:

(P1 ~Pggg *using*
    1. cDNA // <object1>
    2. detecting$_{P2}$ // <purpose>
)
(P2 ~Pgdti *detect*
    1. differential expression of nucleic acid // <object1>

2. sample // <place>

)[2], where the case-role <purpose> of the predicate *using* is filled by the reference to another predicate construction (the reference is shown as a lemmatized form of the predicate and its unique number P2 as a subscript) singled out at the stage as a result of recursive applying the rules of the previous stage to the text filler of the case-role.

7) *Defining component zones content*: depending on its function the predicate may belong to one of three groups: 1) method purpose, 2) method component, and 3) relation. The judgement about its belonging to one of them is made by location principle, i.e. by the position of the predicate in the text. *Component zone* is formed by 1) a single predicate of one of the first two groups (a *forming* predicate), 2) predicates having common arguments with the forming predicate of the zone. For the example, there formed three component zones, including the following predicates (forming predicates are italicised):

PZone1=[*using*, detect]

PZone2=[*hybridizing*, forming]

PZone3=[*comparing*, indicates]

The distribution of predicates of the relation group on component zones may be obviously overlapping. Each predicate construction is assigned labels conveying what group and zones the predicate belongs to.

The output of such indexing is a representation which mirrors predicate-argument structure of a patent claim clustered in compliance with predicate's belonging to a certain group and zone and further used as a unit in the search strategy.

3.2. Query Formulation and Evaluation

We consider a sophisticated search by applying a structured representation of the document content. Search strategies are normally organized as matching functions used to retrieve documents by mapping the description of the query on the descriptions of the documents, at that the representations of the query and documents mostly necessitate to be alike. Following this principle, the query description within the present framework should be a frame-structured predicate construction having arguments marked by its case-roles.

To obtain such description an interface is utilized which suggests a user to input the description of the sought document (invention description) by sequential definition of 1) method purpose, 2) its components and 3) the relations, in terms of predicate construction. At every stage the user is suggested: 1) to choose an ontological concept denoting the action referred to a predicate in the lexicon, 2) define the participants of the situation (arguments) by filling slots corresponding to the case-roles of the predicate. Defining participants may take a form of a recursive procedure if a participant is a relation or an action normally expressed in the language by a predicate. Thus, if a user would like to find a patent for treating a certain disease by the intake of a given medicine, query formulation will be a two-stage process: 1) defining a concept for the treatment; filling the slot <object1> with the term denoting the disease; 2) filling the slot <means> by the concept, denoting the idea of intake (referring to the predicate *administering* in the lexicon), and recursively filling a new frame structure corresponding to the predicate *administering*:

(P1 *treating*

1. bone disease // <object1>
2. $\text{administering}_{P2}$ // <means>

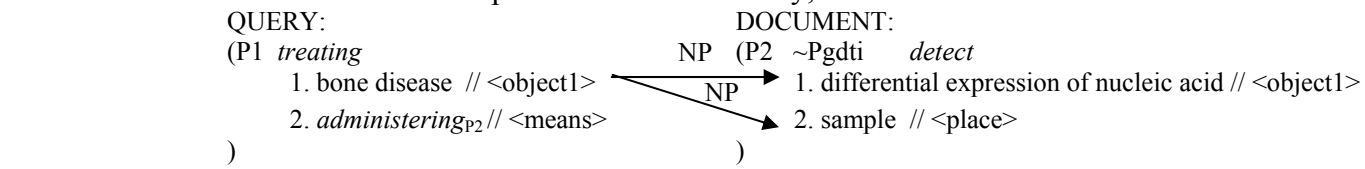
[2] P1, P2 – ID numbers of predicates; ~Pggg, ~Pgdti – supertags associated with the predicates

)
(P2 *administering*
1. powdered sodium alendronate monohydrate // <object1>
2. human patient // <object2>
)

Like a document description, query representation is organized into component zones made automatically which is possible as: forming predicates are inputted at an explicitly defined stage of query specifying; the restriction imposed on relational predicates stipulates that the predicate constructions must comprise at least one of previously defined arguments of another predicate.

On formulating a query, the search strategy is executed through a multi-level procedure. Three levels are taken into account: 1) document/query; 2) component zone; 3) predicate construction. Through the "down-passage" from the most general to the most specific level candidates for similarity to query representation units are picked out from the document representation units of the appropriate level. In the course of "up-passage" a single candidate for the unit at each level is selected on the basis of similarity metrics applied.

Mapping predicate-argument structure (matching at the predicate construction level) takes the central position in the search strategy. The candidates form the document are picked out at the level on the basis of the lexical semantic similarity of predicates, at that the distinction "forming–non-forming" predicate is vital (thus, candidates for forming predicates of a query are searched only among forming predicates of a document). For each candidate the argument content is estimated then. Each argument of a query predicate construction is mapped against every argument (belonging to the same type – noun phrase, adverb, nested predicate construction) of the document candidate predicate. Schematically,

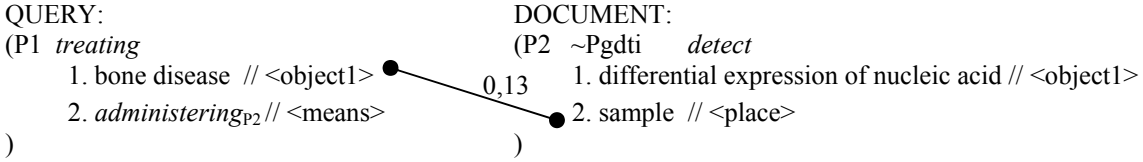


If a predicate has embedded predicate arguments, the corresponding predicate constructions are recursively matched. The most of argument fillers are noun phrases, therefore they are given special attention. When mapping noun-phrase fillers, three aspects are considered to contribute (to a variable extent) in their similarity coefficient: 1) ontological similarity of heads, 2) semantic similarity of the head modifiers, and 3) the identity of the case-roles. Ontological similarity of heads is defined basing on the distance to the least upper bound of the nodes corresponding to the concepts presented by NP-heads. Modifiers' similarity is likewise assessed, allowing for the importance for the similarity of each of them in the overall concept instantiated in a patent text. The identity of case-roles is estimated by means of arbitrary coefficients. Taking it into account, the best candidate is selected for each query argument, basing on the metric calculated as:

$$Term_j = \max_{k \in NPCount_d} \left\{ \sum_{i=1}^{3} Param[i] \times w(Param[i]) \right\}, \; j = \overline{1..NPCount_q},$$

where $NPCount_d$ is the number of NPs in a document predicate construction, $NPCount_q$ is the number of NPs in a query predicate construction, w(Param[i]) – the weight of the i-th parameter, $Param = \left\lfloor Head(NP_q, NP_d), Lex(NP_q, NP_d), SemR(NP_q, NP_d) \right\rfloor$.

The arguments having the best matching function value are associated, while the other candidates are discarded. The correspondence is marked by the value of $Term_j$, denoting the degree of similarity for the two arguments. For example,



To evaluate the similarity of the two predicate constructions the metric calculated as a simple product of the integrated arguments similarity coefficients and the semantic similarity of the predicate words coefficient is utilized.

The metrics similar to the one used for assessing the similarity at the predicate construction level are exploited at the higher levels (component zone, document/query), taking as inputs the values of the metrics calculated at the previous levels. The output of the system then is presented as a ranged list of documents sorted by the value of the metrics at the document/query level.

4. Conclusion

Widely-spread system utilizing Boolean search, though intuitively more comprehensible, cannot provide for fine-tuned mapping of the concepts, especially for the domain where processes and actions (not objects) are in focus, as it is in the domain of patents on method. The presented information retrieval model is an attempt of creating a more sophisticated system exploiting the natural language peculiarities for increasing the performance of the system.

The parsing procedure used for indexing currently encompasses most of the grammar productions and many of the restrictions but it is by no means complete. The current state of the model implies utilizing type relations in the ontological representation disregarding other types of relations. Ontology exploited is modelled as a denotational representation of the domain, though it may be fruitful for improving the performance to map the concepts basing on the designating features mapping as well. All this is considered as directions for further investigation.

Bibliography

[Andreasen et al., 2000] Andreasen, Troels, Jørgen Fischer Nilsson, and Hanne Erdman Thomsen. Introduction: The OntoQuery Project. In *H.L. Larsen et al. (eds.) Flexible Query Answering Systems, Recent Advances*. Physica-Verlag, Springer, 2000. Pp. 15-26.

[Fellbaum, 1998] Fellbaum, C. (ed.). WordNet: An Electronic Lexical Database. – MIT Press, Cambridge, MA. 1998. – 423 p.

[Mahesh and Nirenburg, 1995] Mahesh, Kavi, and Sergei Nirenburg. A Situated Ontology for Practical NLP. In *Proceedings of IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing*. Montreal, Canada. August 19-20, 1995. Pp. 1-10.

[Mihalcea and Moldovan, 2000] Mihalcea, Rada, and Dan Moldovan. Semantic Indexing using WordNet Senses. In *Proceedings of ACL Workshop on Recent Advances in Natural Language Processing and Information Retrieval*. Hong Kong, China. October, 2000. Pp. 35-45.

[Miller et al., 1990] Miller, George A., Richard Beckwith, Christiane Fellbaum, Derek Gross, and K. J. Miller. Introduction to WordNet: an Online Lexical Database. In *International Journal of Lexicography*. Vol. 3(4). 1990. Pp. 235-244.

[Sheremetyeva, 1999] Sheremetyeva, S. A Flexible Approach to Multi-Lingual Knowledge Acquisition for NLG. In *Proceedings of the 7th European Workshop on Natural Language Generation / P. St. Dizier (ed.)*. Toulouse, France. May 13-15, 1999. Pp. 106-115.

[Sheremetyeva, 2003] Sheremetyeva, S. Natural Language Analysis of Patent Claims. In *Proceedings of the Workshop on Patent Corpus Processing*. Sapporo, Japan. July 12, 2003. Pp. 66-73.

[Smeaton, 1999] Smeaton, Allan F. Using NLP or NLP Resources for Information Retrieval Tasks. In *Strzałkowski, T. (ed.) Natural Language Information Retrieval*. Kluwer Academic Publishers, 1999. Pp. 99-112.

[Strzałkowski, 1994] Strzałkowski, Tomek. Robust Text Processing in Automated Information Retrieval. In *Proceedings of the Fourth ACL Conference on Applied Natural Language Processing*. Stuttgart, Germany. October 13-15, 1994. Pp. 168-173.

[Surdeanu et al., 2003] Surdeanu, Mihai, Sanda Harabagiu, John Williams, and Paul Aarseth. Using Predicate-Argument Structures for Information Extraction. In *Proceedings of the 41s Meeting of the Association for Computational Linguistics*. Sapporo, Japan. July 7-12, 2003. Pp. 8-15.

[Tesnière, 1959] Tesnière Lucien. Éléments de syntaxe structurale. Paris, Klincksieck. 1959. – xxvi, 670 p.

[Voorhees, 1999] Voorhees, Ellen. Natural language processing and information retrieval. In *M.T. Pazienza (ed.) Information Extraction: Towards Scalable, Adaptable Systems. Lecture notes in Artificial Intelligence*. Vol. 1714. Springer-Verlag, 1999. Pp. 32-48.