Abstract

The aim of this deliverable is to describe the status of Legal Case Study Prototype at the end of the second year of the project. This document introduces the work done by iSOCO engineers and the UAB team, in collaboration with the JSI team. The cooperation between these teams has produced the following results: (i) the evolution of the Expert System from its preliminary version, introducing: TextGarden technology for domain detection, semantic distance technology (developed by iSOCO) for matching, integration plan with other SEKT technology (Visualization, KAON 2, GATE, TextGarden), ontology evolution, measurement plan; (ii) the description of Jurisprudence System scenarios; (iii) Legal Case Study Prototype integration in SEKT (SIP and SEKT technology), (iv) ontology generation and the Legal Case Study (Question Topic Ontology and Judicial Topic Ontology), (v) some effectiveness and efficiency measurements of current Prototype version; (vi) next steps of Prototype.

Keyword list: legal case study, development

WP10 Case study: Intelligent integrated decision support for legal professionals.

Prototype

Contractual date of delivery: 31/12/2005
Actual date of delivery: 31/12/2005
This document is part of a research project partially funded by the IST Programme of the Commission of the European Communities as project number IST-2003-506826.
D10.3.1 /Prototype

Executive Summary

This document presents the evolution of the Legal Case Study in the second year of the SEKT project, and shows how the prototype has evolved from the preliminary design presented in D10.2.1[5].

In this stage the work was focused on starting the integration of the iFAQ\(^1\) system into the overall SEKT architecture. Currently, the system collaborates with several SEKT components: Expert Knowledge System uses domain detection technology (TextGarden) from workpackage 1 to classify questions in legal subdomains, KAON 2 from workpackage 2 for ontology management and GATE from workpackage 3 for natural language processing. A plan showing how the final version will be developed by the end of the project is also provided. This final version has to integrate in SIP (SEKT Integration Platform) and it will include the Jurisprudence Knowledge System, which will use ontology alignment technology from workpackage 4, search & browse technology from workpackage 5, name entity extraction and knowledge generation technology from workpackage 2.

Subsequently, in Chapters 7 and 8, the use of other SEKT developments is detailed. The results of this usage are shown with the integration of the Ontology of Professional Judicial Knowledge (OPJK) into the top ontology PROTON and the description of the generation of two topic ontologies for the Legal Case Study.

OPJK, the ontology developed by the legal case study team, was extracted from the selection of relevant terms from nearly 200 competency questions. SEKT complementary know-how has facilitated the integration of this domain ontology, the OPJK ontology, into PROTON (PROTo Ontology), developed by Ontotext Lab as a light-weight upper-level ontology. The PROTON ontology contains about 300 classes and 100 properties, providing coverage of the general concepts necessary for a wide range of tasks, including semantic annotation, indexing, and retrieval. This integration has allowed not only the reuse of an existing upper ontology, PROTON but also the maintenance of the necessary “professional” trait of OPJK. This integration took place at two stages: OPJK concept into PROTON concept integration and the reuse of existing PROTON relations.

Regarding topic ontology generation, the primary goal of the task was to develop technology for recognizing semantic concepts from the text, specifically from the questions posed by the judges, in order to develop the ability to automatically annotate further questions and judgments. Through that, QTO (Question Topic Ontology) was produced using OntoGen, a tool provided by the Jozef Stefan Institute. The fundamental idea of text categorization is to automatically determine the category (topic) of a piece of text (such as a document, or a natural language question). There are many applications for such technology. For example, if we can detect semantic concepts in text, we can automatically create a semantic description from the text. By detecting the same concept both in the question and in the judgment, we can find a particular judgment to be relevant. To those goals, a JTO (Judgment Topic Ontology) is currently being developed.

\(^1\) iFAQ is Expert Knowledge System in Legal Case Study Prototype
D10.3.1 /Prototype

After that, Chapter 9 introduces some measurements regarding efficiency and effectiveness.

The deliverable concludes with an enumeration of next steps in Chapter 10. Next steps in 2006 are: evaluation of the results obtained from different configurations of the search engine (keywords, enhanced keywords, semantic distance), integration of the current ontologies version in the last version of PROTON, integration with other SEKT technologies (SIP, ontology alignment, search and browse, etc …), planning and run user tests, to improve effectiveness and efficiency, and development of Jurisprudence Knowledge System.

Finally we present some conclusions (Chapter 11) related with the design adopted: (i) based in NLP and Semantic Web technology, (ii) designed to be efficient, extensible, customizable and scalable, (iii) incremental searching, (iv) pluggable search algorithms, etc.

It includes some related appendices with software distribution and unitary tests.
Contents

Executive Summary .................................................................................................................. 3

Contents .................................................................................................................................. 5

Figure List ................................................................................................................................ 6

List of Tables .............................................................................................................................. 7

1 Introduction ............................................................................................................................ 8

2 FAQ System Architecture ...................................................................................................... 9
   2.1 Main Stages ....................................................................................................................... 11
       2.1.1 Ontology Domain Detection ...................................................................................... 11
       2.1.2 Enhanced keyword ................................................................................................... 12
       2.1.3 Ontology/Semantic Distance ..................................................................................... 12

3 Scenario Description .............................................................................................................. 13
   3.1 User Search Scenarios ..................................................................................................... 13
       3.1.1 User question search scenario .................................................................................. 13
   3.2 System scenarios .............................................................................................................. 28
       3.2.1 Search system initialization use case ......................................................................... 28
   3.3 Administrator scenarios .................................................................................................... 30
       3.3.1 Tuning iFAQ System use case .................................................................................. 30

4 Architecture diagrams ......................................................................................................... 33
   4.1 Component diagram ........................................................................................................ 33
   4.2 Classes diagrams ............................................................................................................ 34
       4.2.1 Main Class Diagram ................................................................................................ 35
       4.2.2 Caching class diagram ............................................................................................. 35
       4.2.3 Administration Class Diagram .................................................................................. 36
       4.2.4 FAQ and Domain class diagram ............................................................................. 36
       4.2.5 Search class diagram ............................................................................................... 37
       4.2.6 NLP class diagram .................................................................................................. 38
       4.2.7 Ontology class diagram ........................................................................................... 39
       4.2.8 Persistence class diagram ........................................................................................ 40
   4.3 Deployment diagram ......................................................................................................... 40

5 Jurisprudence System ........................................................................................................... 42
   5.1 Answer Explanation: Documental Support ...................................................................... 42
       5.1.1 Introduction ................................................................................................................ 42
       5.1.2 Current Solutions and Available Resources ............................................................... 42
   5.2 Description of the Scenario .............................................................................................. 44
       5.2.1 F.A.Q vs Judgment semantic matching ...................................................................... 44
       5.2.2 Search and Browse of Jurisprudence ...................................................................... 45

6 SEKT Integration .................................................................................................................... 47

7 Ontology of Professional Judicial Knowledge ...................................................................... 49
   7.1 PROTON Integration ........................................................................................................ 50
       7.1.1 Class Integration ........................................................................................................ 52
       7.1.2 Inherited Relations .................................................................................................... 52

8 Ontology Generation and the Legal Case Study .................................................................. 53
   8.1 Description of the Problem .............................................................................................. 53
   8.2 Data collection and internal structure ............................................................................ 53
       8.2.1 Collecting questions ................................................................................................. 53
       8.2.2 Collecting judgments ............................................................................................... 54
       8.2.3 Structure of judgement ............................................................................................ 54
   8.3 Hand-Made Semantic Structure ..................................................................................... 55
8.3.1 Pre-existing semantic annotations of questions ................................. 55
8.3.2 Pre-existing semantic annotations of judgements ............................. 56
8.4 Application of natural language technologies: lemmatization ............... 60
8.5 Application of semantic web technologies: topic ontologies in RDF ...... 61
8.6 Application of machine learning technologies: topic detection ............ 62
8.7 Using topic detection to evaluate a manually-constructed topic ontology .. 64
  8.7.1 Analysis of keywords ................................................................. 64
  8.7.2 Analysis of misclassifications..................................................... 66
8.8 Recommendations for text preparation .............................................. 68
  8.8.1 Irrelevant Grammatical Categories ............................................. 68
  8.8.2 Words to be preserved in the corpus ......................................... 69
  8.8.3 Domain dependent words .......................................................... 70
8.9 Application of knowledge discovery: semi-automated topic ontologies .... 70
  8.9.1 OntoGen - a tool for semi-automatic ontology construction .......... 71
  8.9.2 Question Topic Ontology .......................................................... 71
  8.9.3 Judgment Topic Ontology ......................................................... 74
8.10 Topic detection software module .................................................... 75
9 Measurements .................................................................................. 76
  9.1 Effectiveness measurements .......................................................... 77
  9.2 Efficiency measurements ................................................................ 78
10 Next Steps ....................................................................................... 80
11 Conclusions .................................................................................... 81
12 Bibliography and references .............................................................. 82

Figure List.

Figure 1.1: Architecture of the Legal Case Study System ............................ 8
Figure 2.1: Multistage adaptive searching system ...................................... 10
Figure 2.2: Ontology Domain Stage. Statistical Approach .......................... 11
Figure 2.3: Ontology Domain Stage. TextGarden Technology Approach .... 11
Figure 2.4: Enhanced Keyword Stage Approach ...................................... 12
Figure 2.5: Ontology/Semantic Distance Stage Approach ......................... 13
Figure 3.1: Legal Case Study Prototype implemented use cases ............... 14
Figure 3.2: User's interface to search FAQs ............................................ 15
Figure 3.3: FAQ search results screen ..................................................... 16
Figure 3.4: Overall search system UML sequence diagram ....................... 18
Figure 3.5: MixedNLP processing example ............................................ 20
Figure 3.6: Ontology domain detection stage UML sequence diagram ....... 22
Figure 3.7: Ontology domain detection stage based in WP1 technology, UML sequence diagram .......................................................... 23
Figure 3.8: Keyword matching stage UML sequence diagram ................... 24
Figure 3.9: Ontology concepts graph path-matching stage UML sequence diagram .......................................................... 26
Figure 3.10: User's FAQ navigation by domain, screen capture ................. 28
Figure 3.11: Search system initialization use case UML sequence diagram .... 29
Figure 3.12: Tuning iFAQ System UML sequence diagram ....................... 31
Figure 3.13: Ontology processing UML sequence diagram ....................... 32
Figure 4.1: Component Diagram ............................................................ 34
Figure 4.2: Main class diagram .............................................................. 35
Figure 4.3: Caching class diagram ........................................................ 35
Figure 4.4: Administration class diagram ................................................ 36
Figure 4.5: FAQ and Domain class diagram ......................................................... 36
Figure 4.6: Search class diagram ............................................................................ 37
Figure 4.7: NLP class diagram .............................................................................. 38
Figure 4.8: Ontology class diagram ...................................................................... 39
Figure 4.9: Persistence class diagram .................................................................. 40
Figure 4.10: Deployment Diagram ......................................................................... 41
Figure 5.1: Online access to jurisprudence database (Laley) .................................. 43
Figure 5.2: Sentence structure and relevant information ......................................... 44
Figure 5.3: F.A.Q. matching to Jurisprudence ....................................................... 45
Figure 5.4: UML use case corresponding to the functionality ................................. 45
Figure 5.5: Search and Browse Functionalities from the Digital Library Case Study (BT) ................................................................. 46
Figure 5.6: Form based search interface for jurisprudence interacting with the search and browse component .............................................................. 46
Figure 6.1: SEKT Integration diagram of Expert System ......................................... 47
Figure 6.2: SEKT Integration diagram of Jurisprudence System ............................ 48
Figure 6.3: SEKT Integration diagram of ontology engineering in Legal Case Study 48
Figure 7.1: OPJK .................................................................................................... 50
Figure 7.2: PROTON (PROto ONtology Modules) ................................................ 51
Figure 8.1: Screenshot of intuitively detected sub-domains, where the domains are directly related to the activity of the judge ................................................. 56
Figure 8.2: Screenshot of the development of the QTO, only 46% of the questions had been used ............................................................................................................. 72
Figure 8.3: Screenshot of the question visualization functionality .......................... 73
Figure 8.4: Screenshot of the development of the QTO, 74% of the questions had been used and the rest of questions were stored within Guardia-root ......................... 73
Figure 8.5: Screenshot of the development of the main QTO topics into sub-topics.. 74
Figure 8.6: Screenshot of the development of JTO ................................................ 75

List of Tables:
Table 6.1: SEKT Technology in Legal Case Study .................................................. 49
Table 9.1: Measurements ....................................................................................... 77
Table 9.2: Same meaning effectiveness tests results ............................................. 78
Table 9.3: Different meaning effectiveness tests results ....................................... 78
Table 9.4: Efficiency without caching .................................................................... 79
Table 9.5: Efficiency with caching ........................................................................ 80
1 Introduction.

The aim of this deliverable is to describe the status of the legal case study prototype. The approach followed has been to develop an application where the main effort has been centred in user interaction and knowledge base developments.

A previous version of the prototype was used last September at a User Validation Workshop in Barcelona. The aim of this meeting was to define and plan work with a set of end users in order to validate the user needs and the functionalities of the system (the meeting was organized in collaboration with WP8 - specifically work on the usability, optimisation of business benefits and benchmarking. A previous version of the prototype was presented, with a set of measurements, at the Q7 Meeting in Madrid (October 2005). As a result of these meetings we captured some more requirements and identified some new applications of the SEKT technology.

In this deliverable, we describe in detail the first part of the prototype (current status, developed requirements, architecture, integrated SEKT technology, next steps, etc). This part is related to the development of the FAQ searching system or Expert Knowledge System (shown on the left hand side of the Figure 1.1.). The aim of this system is to find the best match between a question posed by a user (usually a new judge) and the expert knowledge of the judges stored in the form of a set of question-answer pairs. In order to find the question-answer that best matches the input question, in terms of semantic similarity, the following technologies are applied:

- Natural language analysis and processing - to link the terms in the sentence to the ontology.
- Calculation of the semantic distance between the set of terms contained in the input sentence and the set of terms contained in the stored questions (this process is ontology-based).
Regarding the second part of the deliverable (Jurisprudence System, on the right hand of Figure 1.1), we describe the approach that was followed to create an ontology automatically from the judgments stored in a database. We also present current solutions and available resources related with judgments. The scenarios for Jurisprudence Knowledge Application are described in section 5: (i) FAQ vs Judgement Semantic Matching Search; (ii) Search and browse of judgments.

The document is divided into eleven main sections. Sections 2, 3, and 4 are related to the Expert System. Section 2 comprises a description of the architecture and the design approach that was taken when designing the final system. End user scenarios for the application are described in section 3. The main classes and the sequence diagrams are introduced in section 4. The Jurisprudence System is described in section 5. The integration of the prototype with SEKT technology and architecture is described in section 6. Section 7 comprises the current version of the ontology of professional judicial knowledge and its integration with PROTON\(^2\). Ontology generation in Legal Case Study is described in section 8 (collaborative work with JSI, WP 1). Section 9 describes some results concerning the application of effectiveness and efficiency measurements on the Legal Case Study prototype. Section 10 describes the development of the prototype in the final year of the project. Finally, we close with some conclusions about the prototype system (section 11).

2 FAQ System Architecture.

Our improved FAQ searching system design is based on Spanish NLP, background calculation of ontologies, caching of background calculated data, and a multistage searching approach with progressive delimitation of the FAQ target.

The software design takes into account the interaction of the user with the system. A user types a question and expects the system to find a FAQ candidate as close as possible to the question. To achieve this goal, several search and score algorithms have been designed based on Natural Language Processing and on Ontology Concepts Matching [26]. Algorithms have been organized around an architecture based on an adaptive multistage search chain, which is based on a variation of the “chain of responsibility” pattern:

- The search process is subdivided into several cooperative stages.
- At each stage, the system applies an algorithm that reduces the incoming target FAQs, producing a narrower version of that set.
- Each search stage is premised on the belief that the FAQ’s question that best fits the question posed by the user belongs to the outcome set of FAQ’s at that stage.
- The multistage search algorithm allows the system to stop when the FAQ target has been reduced considerably (beyond a parameterized threshold). Therefore, the stages are not necessarily exhausted, and the computational cost is reduced and adapted to the search needs.
- Besides, multistage searching allows the system to show the best score results to the user before completing all available search stages. In this way, users can

\(^2\) http://proton.semanticweb.org/
D10.3.1 /Prototype

decide whether to accept the result or continue with the remaining search stage.

The design of the system is based on the following technological considerations:

- **Accurate search.** This system will be able to find the best possible matching FAQ question with the user’s question.
- **Efficiency.** Searching must not take a long time to find a result successfully, and must scale well with the FAQ repository size.

The key technologies used in the architecture, which enable these design considerations to be met, are:

- **Natural Language Processing.** NLP is used at several search stages to acquire additional comprehension from the user’s question. A morphological analysis of the user’s question is performed. The relevant words and grammatical patterns drawn from the question are used by components in other stages.
- **Thesaurus Processing.** Thesaurus processing is used to match words based on synonym relationships. The system attempts both exact and synonym matching.
- **Ontology Processing.** The system makes use of ontologies to obtain additional understanding of the user’s question. The system attempts to find out a track of the user’s question, or part of it, in the ontology by building a graph path that is compared to each of the stored FAQ graph paths.
- **Cache proxy.** To avoid the repetition of computations, the system produces intermediate results of repetitive calculations that can be saved. Many of these calculations can also be recovered from a repository like a RDBMS and saved in cached memory.

![Multistage adaptive searching system.](image-url)

Each stage is described in the following subsections.

---

3 All the processes made over a question input are also made over all stored question. These results are also stored in a relational database.
2.1 Main Stages.

2.1.1 Ontology Domain Detection.

The main purpose of this stage, as described in deliverable D10.2.1 [5], is to determine the FAQ domain target set, based on user question analysis, for usage at a later stage. The complete FAQ database is made up of over 35 legal subdomains. The first search stage focuses on reducing the FAQ domain target set, selecting the related FAQs of a legal subdomain. It determines the legal subdomain of a user question using two different approaches. The first approach is based on a statistical recount of occurrences of all relevant concepts of the user question among the different domain ontologies (see Figure 2.2). The second approach is based on the question classification technology provided by WP1 (see Figure 2.3) that is based on the technology of TextGarden\(^4\). The Legal Case prototype can be configured to use one of these approaches as part of the multistage search engine.

---

\(^4\) [http://kt.ijs.si/Dunja/textgarden](http://kt.ijs.si/Dunja/textgarden)
2.1.2 Enhanced keyword

At this stage, the user question is word-like tokenized and system tries to match each token with each FAQ question token. “Exact”, “synonymy” and “morphological” matching are tried. As a result of this stage, a narrower list of candidate FAQ items will be supplied. Although this might seem similar to other standard searching systems using keywords or metadata, here the difference lies in the use of morphological parsing of the user question that discards non-relevant words, and the use of synonymy and morphology for matching (described in detail in deliverable 3.1.1).

![Enhanced Keyword Stage Approach](image)

**Figure 2.4:** Enhanced Keyword Stage Approach

2.1.3 Ontology/Semantic Distance

The calculation of ontology/semantic distance[9] is very time consuming, and therefore it has been left to the final stage, after the FAQ target database has already been reduced considerably (by the filter of the previous searching stages). Grammatical patterns are detected from user question at this stage. These patterns are then searched for in the Ontology in order to build a graph path or trace. Finally, the system tries to match the user question ontology graph path with a reduced FAQ target subset of graph paths that have been calculated in the background previously, using the semantic distance algorithms.

This stage offers customization in:
- Function of semantic distance between ontology nodes.
- Heuristic for semantic graphs construction.
- Function of semantic distance between ontology graphs.
3 Scenario Description.

The initial use cases designed for the FAQ searching system are described in detail in [5]. This document describes the final version of these scenarios, including some related details of their implementation (screen captures, partial results of the phrase processing, etc).

The use case UML diagram shown in Figure 3.1 depicts the most important cases, those that have been implemented. The main changes that have been introduced are related to the iFAQ administrator’s tasks (shaded use cases are either new or updated in the development).

3.1 User Search Scenarios

3.1.1 User question search scenario

Intention of scenario/setting: This is the most frequent and important use case found in our system. A user types a question that he/she expects to find in a FAQ repository. The system determines the question that has meaning most similar to the question input by the user (the FAQ questions are stored in the Expert Knowledge System database). The system will use technologies available in SEKT project to try to find the FAQ item with the best score.

User expectations: The user expects to find at the FAQ repository the question/answer that can answer his/her question, i.e. the question stored at the FAQ repository that has a meaning that is most similar to the question input by the user. The user expects a reasonable response time (less than 2 seconds).

Pre-conditions: The software design takes into account the following functional considerations:

- Exhaustive search can be very time consuming.
- Multistage searching system lets the system stop when the FAQ target set has been reduced considerably. Therefore, not all stages are exhausted and the
computational cost is reduced and adapted to search features. We introduce a threshold value related with the result list size, so system interrupts the search, when the size of partial result list is equal or smaller than this threshold value.

- Multistage searching allows the system to be configured to show the best score results to the user before all available searching stages are completed. Users can choose to accept the result or to continue with the remaining search stages.

**Figure 3.1:** Legal Case Study Prototype implemented use cases.

**Post-conditions:** The system processes the user’s questions and returns the best FAQ list related to the question with their scoring (the scoring is related to the degree of matching between the question-input and the question stored [8]).

**Alternative scenarios:** Related use cases are *Overall search system, Ontology domain detection stage, Keyword matching stage* and *Ontology concept graph path matching stage*. These use cases are described in detail in the next sections.

**Screen captures:** The following figures show screen captures of the user's search. The interface to the iFAQ system is shown in Figure 3.2. The related results screen is shown in Figure 3.3. This screen contains three main boxes:
D10.3.1 /Prototype

- “Pregunta formulada” that contains user question with its highlighted relevant words.
- “Pregunta encontrada” that contains the question of the first FAQ in the result list. It also includes the matching between user question and FAQ question relevant words.
- “Respuesta” that contains the response of this FAQ.

We also include a button (“Ver otras preguntas relacionadas”) to see other FAQ of the result list. The box “Satisfecho” allows the user to communicate its satisfaction degree with the system response.

Figure 3.2: User's interface to search FAQs.
In Figure 3.4, we show the UML sequence diagram for the multistage search system. The software architecture uses a Factory pattern to build a FAQSearchEngine implementation suitable for ontology search. In our case, ontology based FAQSearchEngine is created. Other FAQSearchEngine could be used if necessary. FAQSearchEngine will determine, either from configuration or on demand, which search engine to use from those available.

Three further stages of processing reduce the FAQ target set, with the compromise that the best scoring FAQ item related to the user question must be included in this set. This last point is very important because if previous searching stages determine a FAQ target set incorrectly, the next searching stages will be unprofitable, because in this case, system applies a search process over a FAQ set, that not contains the FAQ target.

The input to each searching stage will be a FAQ subset that was determined at a previous stage. The output will be a reduced FAQ subset from the narrower search. We postulate two assertions in our design:

- The FAQ item that best matches the user’s question (FAQ target) must be included in the output FAQ subset.
- Each successive stage is more restrictive than the previous one, i.e. the FAQ result set narrows at each stage.
The three searching stages have been designed to leverage the following ontology and NLP technologies: ontology domain detection, keyword/synonym detection and Ontology concept graph path matching (as we have described at section 2).

Another technology we will use frequently is a cache system that gives rapid access to frequently used data, such as, for example, the FAQ database. Performance is the main motivation for using this cache-based design. Some data, which the search system makes intense use of, does not need to be recalculated more than once. As cache memory is limited, a compromise between memory consumption and efficiency has to be managed. A core manager (SemanticDistanceManager) allows managing, in general, all factories of the architecture, in particular the search engine factory.

As a result, a FAQ list (ordered by score) is provided from a completed searching process (it is the final result of the search engine).
Figure 3.4: Overall search system UML sequence diagram
Ontology domain detection stage

The next UML sequence diagram (Figure 3.6) depicts in detail the workflow followed at this stage, our first searching stage. Its goal is to reduce the FAQ database target of our searching system in order to improve performance in the other phases. The FAQ database can be very vast. Indeed, the FAQ database is composed of several smaller domain databases. As the user’s question is likely to belong to a knowledge domain, it is only necessary to search in that specific domain database, not in the remaining databases. The sequence diagram shows that the searching system is built over some important components that will be used often:

- **NLPEngine**: responsible for morphologic and grammatical parsing of questions.
- **MorphologicalEngineAdapter**: a helper class used by NLPEngine, specialized on morphological analysis. This adapter leverages on specialized external morphological analyzers supplied by other companies.
- **ThesaurusEngineAdapter**: a helper class used by NLPEngine, specialized on synonyms searching. This adapter leverages on specialized external thesaurus engines supplied by other software manufactures.

The design, based on adapters, allows us to replace those specialized helper classes with alternatives, provided that these new helper classes fulfil the adapter interface. Adapters allow us to use plug-ins for special tasks.

We developed a combined NLP and Thesaurus Engine, named MixedNLPEngine. It dissects the user question and detects all its relevant words, analyzing them morphologically. The combined engine also detects the concepts, attributes or instances of the ontology that are in this list of relevant words (ontology linking). If the relevant words do not have a correspondence with the ontology, this search engine tries to find the correspondence with the ontology using synonyms.

The next example shows the result generated by MixedNLPEngine when processing a phrase entered by a user:

![User Phrase](image)

<table>
<thead>
<tr>
<th>Word</th>
<th>Synonyms</th>
<th>Morphological</th>
<th>Ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>venido</td>
<td></td>
<td>• venir</td>
<td></td>
</tr>
<tr>
<td>mañana</td>
<td></td>
<td>• mañana</td>
<td>Classes: <a href="http://protege.stanford.edu/kb#Mujer">http://protege.stanford.edu/kb#Mujer</a></td>
</tr>
<tr>
<td>señora</td>
<td>Mujer</td>
<td>• señor</td>
<td></td>
</tr>
</tbody>
</table>

5 One lady came this morning in order to get “protection warrant”. We have been busy the whole day with this subject. I’m giving her the “protection warrant” but now she tells that wants to retract the “denuncia” but she doesn’t want the “protection warrant” anymore. What shall I do?
<table>
<thead>
<tr>
<th>Nupla</th>
<th>Slots</th>
<th>Classes</th>
<th>Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>quería</td>
<td>• querer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>orden de</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>protección</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Llevamos</td>
<td>• llevar</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• llevarse</td>
<td></td>
<td></td>
</tr>
<tr>
<td>día</td>
<td>• día</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tema</td>
<td>• tema</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• temer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>acabó</td>
<td>• acabar</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• acabarse</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• acabó</td>
<td></td>
<td></td>
</tr>
<tr>
<td>acordar</td>
<td>• Acordar</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• acordarse</td>
<td></td>
<td></td>
</tr>
<tr>
<td>notificando</td>
<td>• notificar</td>
<td></td>
<td></td>
</tr>
<tr>
<td>orden de</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>protección</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dice</td>
<td>• decir</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• decirse</td>
<td></td>
<td></td>
</tr>
<tr>
<td>retirar</td>
<td>• retirar</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• retirarse</td>
<td></td>
<td></td>
</tr>
<tr>
<td>denuncia</td>
<td>• denuncia</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• denunciar</td>
<td></td>
<td></td>
</tr>
<tr>
<td>no</td>
<td>• no</td>
<td></td>
<td></td>
</tr>
<tr>
<td>quiere</td>
<td>• querer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• quererse</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.5: MixedNLP processing example
For each task mentioned above, the engine leverages other helper classes, those that offer special functions. A morphological engine adapter processes each word in a user question, detecting the POS, gender, number, etc. of each meaning of the word. All relevant words are filtered\(^6\). This task is very computational consuming. Therefore, the most frequently processed words are cached in memory. Later accesses to that data are collected from the cache memory (instead of morphologically processing the user question again).

At this stage, we have all the data, processed from the user question, needed to find out the best candidate ontology domain that matches the user’s question. The next task is to determine the domain of the question input by the user. We can use two alternative approaches to achieve this task:

- If we have several domain ontologies, we can statistically detect all those concepts in each ontology domain. We implement a solution based on the number of occurrences of the same ontology term in the phrase. In the actual prototype we use an ontology API adapter (\texttt{KPOntologyAdapter}), which is based on a KPOntology API (iSOCO technology), to access the ontology. We integrated KAON 2 as an extension of KPOntology, and in the future we intend to integrate KAON 2 as a pipelet of SIP (SEKT Integration Platform).

  To avoid repeated searches of the ontology nodes, these can be recovered from cache (if the performance of the ontology searching system is not as expected). The outcome is an ontology domain candidate with the best statistical score.

Once the ontology domain has been determined, the stage’s final output is built. The output consists of the complete FAQ list of the candidate ontology domain.

Another aspect of the design that was considered was that all data calculated at one stage must be kept for the complete multistage cycle as it is likely to be reused in the following stages. For example, the relevant words meanings of a user question are kept in cache for use later.

- The second approach is based on domain detection using the TextGarden technology (technology provided by workpackage WP1). Currently, all stored question-answer are manually classified. This classification serves as an input to this software so that it can learn about how to make the classification, and also to provide a new classification of this input. Besides, the result of this process can be used to detect the domain of the input question.

The next sequence diagram (Figure 3.7) shows an alternative solution based on the technology of work package 1. We can see that the open architecture allows the implementation of the stage to be changed without collateral effects.

\(^{6}\) The applied filter entails considering as relevant words, verbs, nouns, adjectives, adverbs and pronouns
Figure 3.6: Ontology domain detection stage UML sequence diagram
Figure 3.7: Ontology domain detection stage based in WP1 technology, UML sequence diagram

Keyword matching stage

The second searching stage (Figure 3.8) receives a FAQ list processed in the previous stage. The purpose of this stage is to filter the input to reject non-matching FAQ subset, considering enhanced keyword matching. This stage punctuates each FAQ item with a score. All FAQ items with scores below a set threshold will be eliminated from the candidate list. At the moment, this score is a relation between the questions words, which match (exact, synonymy or morphological matching) and the total number of words.

The key concept at this stage, which differentiates it from other standard search systems, is the use of morphological meanings and synonyms in the keyword-matching algorithm. This avoids the use of exact or partial word matching. The use of only relevant words reduces the emergence of numerous false candidates.

After processing or caching the relevant words of the user question and its synonyms using an NLPEngine (MixedNLPEngine), the searching stage pursues iterating over each input filtered FAQ item. For each FAQ item, the following strategy is used to determine a score giving the degree of the match between a user question and an FAQ question: first, the relevant words of the FAQ question are recovered from the cache (otherwise, they will be recovered from persistence RDBMS\(^7\)). Then, two inner loops iterate along all the relevant words of the user’s question and the FAQ question. For each couple of meanings, an exact match is tried and if it fails, synonymy matching and morphological matching are tried.

\(^7\) Relational database management system. Currently we use mysql.
Figure 3.8: Keyword matching stage UML sequence diagram.
Finally, all collected hits that match are processed to assign a final score to that FAQ item. As a result, we have a list of punctuated FAQ items. Each FAQ item with a score below a threshold is discarded.

**Ontology concept graph path matching stage**

The final stage is the most time consuming (computationally), so it is important to begin with a much reduced input FAQ candidate list. The key steps at this stage can be summarized as follows:

- Parsing of the user question to detect grammatical patterns.  
- Identifying these grammatical patterns in the ontology, implying:
  - Searching for the concepts of the grammatical patterns.
  - Finding the paths that connect those concepts.
- Computing the minimum distance between grammatical pattern paths of user question and FAQ question, which imply computing the minimum distance between two nodes of the ontology.

The next UML sequence diagram (Figure 3.9) depicts this stage in detail.

After mixed NLP processing of the user question, we have a related list of ontology nodes. The next step consists of searching grammatical patterns of these nodes in the domain ontology. This is a complex process consisting of the following different steps:

- Ontology linking: Ontology nodes that match any grammatical concept of the user’s question.
- To obtain the minimum distance path that connects each pair of consecutive concepts (semantic distance heuristic).
- To obtain related ontology graphs: building a minimum path by connecting some or all the pattern concepts in the correct order (semantic graphs construction heuristic).

The first task consists of locating the ontology nodes corresponding to each concept of the grammatical pattern in the correct order (ontology linking). The system can locate these nodes from the cache (the most frequent used nodes), or by searching them with the help of an ontology API framework. We have defined a mixed engine that detects in the user phrase the ontology terms from the NLP.

A connecting path is built as follows: from each node, the system will find a minimum path to the next node, and so on. By doing so, the path that connects all nodes will be completed. In cases where it is not possible to connect a pair of nodes, the resulting path will not be fully connected.

---

8 A typical grammatical pattern is formed by a subject, a verb and some direct or indirect objects. Other patterns may be more or less complex. This syntactical analysis is useful to organize the concepts on the basis of the role they played in the sentence.

9 Currently these searching of grammatical patterns are based on the detection of the relevant words, being a grammatical pattern the sequence of relevant words. This is a temporally solution to avoid the lack of not to have a minimal Syntactical analyzer, that allows us to identify syntactical patterns like the noun phrase, the verb phrase etc.
Figure 3.9: Ontology concepts graph path-matching stage UML sequence diagram.
Now, we face the challenge of finding a minimum distance path between two nodes. A minimum distance path between two ontology concepts is that with the smallest accumulated weight, considering that between two connected nodes there is an edge with a slot weight assigned. We have opted for a variant of Dijkstra’s algorithm. This variant is proposed in the dynamic programming paradigm using binary heaps (developed with structures of Java Framework Collections\(^\text{10}\)) that reduces the algorithm’s running time from $\Theta(n^2)$ to $\Theta(m + n \log n)$, where $n$ stands for the number of nodes, and $m$ for the number of edges. This algorithm uses data representation structures, like adjacency matrix and adjacency lists, which improve its efficiency. Adjacency structures are previously calculated during the system initialization phase and are recovered from cache (in other words, we make the data persistent).

Calculation of the minimum distance path between two ontology nodes is very time consuming. Therefore, for performance reasons it is desirable to keep as many paths as possible in the cache, so they are previously calculated during the tuning phase and are recovered from the cache.

These steps are configurable. The iFAQ system allows the semantic distance heuristic to be defined (e.g. distance between instances, distances between slot attributes, distance between semantic graphs, etc). The semantic graphs construction heuristic (the heuristic that defines how the system generates pattern graphs from paths of ontology nodes) can also be defined.

As a result of this stage, we get a list of ontology paths for each grammatical pattern detected on the user question. Also from the cache, we have similar ontology paths for all FAQ grammatical patterns. Summarizing, we have acquired all the data needed for the next step.

**FAQ Navigation by domain**

**Intention of scenario/setting:** This scenario describes how a user navigates the FAQ repository by domain. The user selects this option from the menu. After that, the system shows a list of FAQs grouped by domain. The user can then access the content of its FAQs.

**User expectations:** The user expects to navigate the FAQ repository by domain. The user is given access to a set of FAQ, grouped by their domain.

**Pre-conditions:** Imported FAQs are classified by domain.

**Post-conditions:** FAQs grouped by domain.

**Alternative scenarios:** The “shows FAQ detail” scenario is included in the current scenario, because when the user selects some FAQ detail, the system must execute the “shows FAQ detail” scenario.

---

\(^{10}\) The Collections Framework. Copyright © 2002 Sun Microsystems, Inc. All Rights Reserved. [http://java.sun.com/j2se/1.4.2/docs/guide/collections/](http://java.sun.com/j2se/1.4.2/docs/guide/collections/)
Screen captures: The next figure (Figure 3.10) shows application screen captures of the user's FAQ navigation by domain.

3.2 System scenarios

3.2.1 Search system initialization use case

Intention of scenario/setting: The search system leverages on some preliminary calculations made at start-up, or in the background during normal working, or with a schedule of low system activity.

When the system starts up, it must execute some initialization tasks. The following UML sequence diagrams (Figure 3.11) summarize this task.
Figure 3.11: Search system initialization use case UML sequence diagram
At start-up, the system creates and configures the main manager, SemanticDistanceManager. Main manager start-up creates and configures all system factories and search engines.

In the particular case of an NLPOntologyFAQSearchEngine, the engine we use in this prototype, the initialization stage consists of the following steps:

- Create and configure Persistence Engine and Cache Repository.
- Create and configure Faq Repository, Administration Tools, Ontology Repository, NLP Morphological Word Repository and Ontology Distance Repository.
- Create and configure the FAQSearchFactory and use it to select NLPOntologyFAQSearchEngine as a search engine.

The system loads the stored data in their related cache (ontologies, semantic distances, faqs, etc.) in the start-up of all repositories (cacheable repositories).

**User expectations:** The computation time associated with the user’s search should be minimal.

**Pre-conditions:** search process, and data to cache are correct.

**Post-conditions:** Ontologies are cached, so minimum computation time in user’s search.

### 3.3 Administrator scenarios

#### 3.3.1 Tuning iFAQ System use case

**Intention of scenario/setting:** This scenario and its alternative use cases (New FAQ item acquisition use case Existing FAQ item updating use case, Ontology processing use case and Topics Ontology Visualization) are maintenance purposes (Figure 3.12). Normally a legal domain expert revises both the FAQ repository and domain ontology. As a consequence, the FAQ searching system evolves with time, incorporating new FAQ items or updating some of them. Also, we have to consider the updating of the ontology (its modification or extension), because it implies the updating of the results of the initialization phase. For this reason, this scenario needs to be incorporated in the system. The administrator uses this scenario to maintain the system updated, because changes in the ontology imply to repeat the FAQ pre-processing.

In order to help administration users to understand the ontology of topics (which it is used to classify FAQs in domains), and select the topics (domains) of a FAQ, when it is added, we will include Visualization technology from WP5 in the administration scenarios. The Topics Ontology scenario explains in detail the visualization of the topics ontology.

**User expectations:** The system must be maintained and updated.
D10.3.1 /Prototype

**Pre-conditions:** Legal domain expert revises both the FAQ repository and the domain ontology.

**Post-conditions:** The system is maintained and updated.

**Alternative use cases:**
*New FAQ item acquisition use case, Existing FAQ item updating use case, and Ontology processing use case.*

![Figure 3.12: Tuning iFAQ System UML sequence diagram](image)

**New FAQ item acquisition use case**

The case when a new FAQ item needs to be added to the repository is described in section 5 of deliverable D.10.2.1 [5].

**Existing FAQ item updating use case**

The case when a FAQ item needs to be updated is described in section 5 of deliverable D.10.2.1 [5].

**Topics Ontology Visualization**

**Intention of scenario/setting:** The objective of this scenario is to provide a tool, which can be used by administration users to visualize and understand the ontology of FAQ domains (topics ontology). This tool is a visualization applet developed by WP5, which can be introduced in FAQ management to make a 3D representation of the ontology.

**User expectations:** Administration users can understand better the ontology of topics.
Pre-conditions: Legal domain expert revises the topics ontology.

Post-conditions: The system is maintained with a FAQ domain (topics) classification.

Ontology processing use case

One of the most important considerations in the FAQ system searching design is the efficiency, and computation of the minimum distance path between two ontology nodes. This is very time consuming. Therefore, for performance reasons it would be desirable to keep as many minimum distance paths as possible in the cache, so they are previously calculated during the tuning phase and are recovered from the cache, in the tuning phase (Figure 3.12).

The following UML sequence diagram (Figure 3.13) shows how ontology processing computes and stores all the distances between nodes of the ontology, using an adaptation of Dijkstra’s algorithm, using Java heaps.

Figure 3.13: Ontology processing UML sequence diagram

Ontology updating use case

Intention of scenario/setting: This scenario includes the previous scenario. In short, when a domain expert has modified the ontology, all FAQ items need to be preprocessed again to update their internal data structures and the ontologies need to be processed. The system needs to be re-tuned.

Finally, all recalculated data is updated both in the repository database and in cached memory.

User expectations: When a domain expert has modified the ontology, the system needs to be re-tuned.

Pre-conditions: A domain expert has modified the ontology.

Post-conditions: The system is maintained/updated.
Alternative scenarios: Tuning iFAQ System use case

4 Architecture diagrams

This section presents the group of components, which build up the complete search system. After that, we present some class diagrams that depict the main classes and their relationships with each one. Finally, the last sub-section includes a deployment diagram, which shows how the components are distributed in the hardware architecture.

4.1 Component diagram

The main subsystems (see Figure 4.1) are:

- NLPSubsystem.
- OntologySubsystem.
- Business Logic, which includes different search engines.
- Cache Subsystem.

Our search system can offer its services to its clients as long as they send a request. Among them, we enumerate the following clients:

- Webservice: the client web service connects to our system to request some search services.
- Web client: this client can be connected directly to our system to request some search services, or it can invoke the related web service.
4.2 Classes diagrams

As part of our design, we have built some class diagrams that depict the main classes and their relationships with the components shown in the previous diagram (Figure 4.1).
4.2.1 Main Class Diagram

![Main Class Diagram](image)

**Figure 4.2: Main class diagram**

4.2.2 Caching class diagram

![Caching class diagram](image)

**Figure 4.3: Caching class diagram**

35
4.2.3 Administration Class Diagram

![Administration Class Diagram](image1)

Figure 4.4: Administration class diagram

4.2.4 FAQ and Domain class diagram.

![FAQ and Domain Class Diagram](image2)

Figure 4.5: FAQ and Domain class diagram
4.2.5 Search class diagram

Figure 4.6: Search class diagram
4.2.6 NLP class diagram

Figure 4.7: NLP class diagram
4.2.7 Ontology class diagram

![Ontology class diagram]

Figure 4.8: Ontology class diagram
4.2.8 Persistence class diagram

Figure 4.9: Persistence class diagram

4.3 Deployment diagram

Considerations of scalability, efficiency and memory limitation have led us to consider a distributed architecture, with different nodes being used to process the user’s search request:

- One or more nodes for NLP processing.
- One or more nodes for Ontology processing.
- One node to gather user requests as a main dispatcher.

All nodes publish their services with web services public interface. All connections between nodes use this architecture. This allows us to improve the system efficiency in terms of global memory and processing, so that it fits itself to increasing demand of services. It is important to note that NLP and especially ontology processing tend to consume a lot of memory.

This architecture is shown in Figure 4.10.
Figure 4.10: Deployment Diagram
5 Jurisprudence System.

5.1 Answer Explanation: Documental Support

5.1.1 Introduction

The legal profession has suffered a dramatic growth during the last thirty years both in USA and the EU\(^{11}\), anticipating what is known at present as “legal global pluralism”. Law, legal systems and organizations have been subject to fundamental changes over the last decades, adding complexity to the legal fields. On the one hand, lawyering, sentencing, and legal drafting have been increasingly growing. In the USA and in many countries alike, there are three to four times more lawyers and legal professionals than in 1950. On the other hand, the production of legal statutes, codes, rulings, etc., has followed the same inflationary path. This has led to a situation in which two of the main problems are handling the complexity and types of legal knowledge, and having reasonable ways to store, retrieve and structure a great amount of legal information.

The legal sector is inherently a knowledge-intensive profession that can significantly benefit from the application of Semantic technologies (see the recent book on Law and the Semantic Web\(^{12}\)). It is also a very challenging sector, both from a technical point of view (quality of required results, complexity of the information involved, variety of content types) as from a content point of view (different -but converging- national laws, different languages, etc.).

As described in the overall case study introduction, judges not only need for highly precise answer to their query but also claim for addition explanation of the offered recommendation coming from actual jurisprudence. For that reason the case study also offers functionalities of enhancing the retrieved answer with existing jurisprudence from known legal databases.

The challenge is to retrieve relevant sentences with the offered answer, in order to help the judge to support any decision to take. Sentences are, some how, containers of legal arguments useful in the decision support. It is crucial for judges to be able to browse through arguments and sentences in order to reuse the jurisprudence.

5.1.2 Current Solutions and Available Resources

The most common software application for lawyers and judges is a jurisprudence database where they can search for relevant documents (sentences, appeals, agreements, bibliography, etc.) when preparing a new case. There are three companies leading the Spanish market: Aranzadi (www.aranzadi.es), Derecho (www.derecho.com) and LaLey (Wolters Kluwert www.laley.net).


Online Jurisprudence Database

Our purpose in this case study is to relate available sentences from Laley with available answers improving the user experience by adding relevant information.

The LaLey application called Nexus allows for key-word based search with advanced meta-information. The most common search criteria are:

- Full text search in sentence summary of header
- Sentence listing by categories (according to proprietary taxonomy)
- Sentence retrieval by key words (according to proprietary vocabulary)
- Search by norms applied in the sentence resolution
- Search by law number applied
- Search by jurisdiction (Civil, Social, etc.)
- Search by court (Supreme, Local, etc.)
- Search by date
- Search by judge name
- Search by resolution type (sentence, appeal, agreement, etc.)
- Search by sentence/appeal number

Most frequent judges complaints are concerned with

- Search precision and recall (many sentences are retrieved and not all of them are relevant to the case the judge is working in)
- Interface: it is difficult, and it takes long time, to construct a query that would retrieve desired sentences.

Sentence Document Structure

Each sentence stored in the data base has a standard document structure. They have several differentiated parts:

- Heading
- Sentence Summary
- Case History
- Decision Rules
- Ruling
Each part of the sentence may contain meaningful information about which rules were applied, who was the judge in charge, dates, arguments, etc.

5.2 Description of the Scenario

This use interaction starts when the user has posed a question to the system, and it has provided an answer. Once the judge gets an answer, she/he feels the need for some more information regarding the decision recommended, and chooses to obtain an explanation for it.

5.2.1 F.A.Q vs Judgment semantic matching

To select the corresponding cases associated to the answer provided, the question and the answer are linked to the corresponding concepts in the Ontology of Professional Legal Knowledge. The set of concepts that represent the question and the answer is then transformed into the corresponding set of concepts that appear in the Jurisprudence ontology. With this set of concepts, which also may appear in any of the databases ontologies, and taking into account, which of them belong to the question and which of them belong to the answer, the cases that are representative are retrieved and presented to the judge.
The relationships between the FAQ response and the jurisprudence are constructed, taking into account the sentence structure, relevant information contained (dates, law numbers, judicial concepts, arguments, proper names, etc.) and the OPJK ontology.

Considerable effort has been expended in defining the relationships between the question-answer pair and the jurisprudence. One of the first steps has been related with the reclassification of the question-answer provided by the judges on the different domains. The result of this work is an ontology named QTO (Question Topic Ontology) obtained as a result of the application of OntoGen (Ontology extraction tool), taking into account a previous manual classification.

The preliminary work on question classification using OntoGen has been used to construct automatically the Judgment Topic Ontology (JTO), again using OntoGen, but with the input being the online jurisprudence described before. This process is described in detail in section 8.

Finally, we can use these ontologies in a future application of mapping technology, provided by the WP4 and described in Figure 1.1.

5.2.2 Search and Browse of Jurisprudence.

Even though the main purpose of the jurisprudence extension is to provide answer explanation and to complete the result of Expert Knowledge System with related judgements, we foresee that the current usage of jurisprudence databases will continue to be used in the system. Judges will also have the opportunity to search for jurisprudence without the need to formulate an explicit question in a FAQ form. For this reason, we will integrate the BT search and browse component (D.5.5.2 [10]) into the jurisprudence system.
The availability of the OPJK incorporating the jurisprudence vocabulary and the JTO to classify the jurisprudence on topics will allow for advanced search and browse functionalities over the legal database. The possibility of performing search on the jurisprudence database will be enhanced using the OPJK ontology.

Judges will formulate search criteria using the ontology navigation. Also retrieval criteria will be based on our semantic distance algorithm. The usage of the ontology will allow for more complex queries than traditional key-word based engines.
6 SEKT Integration

The architecture of our FAQ search system is characterized by being an architecture based on the use of adapters and factories, and by being a heterogeneous architecture in relation to the technologies used (it includes different technologies: Natural Language Processing, Ontology Model, etc).

One of the goals of the Expert System (FAQ searching system) is its integration in SEKT. The following figures (Figure 6.1 and Figure 6.2) show the Expert System and the Jurisprudence System in SEKT Architecture.

Our applications introduce SEKT technology, to improve their behaviour, developing specific implementations of defined adapters. For example, we developed an adapter that uses TextGarden technology to detect the domain of a user question (2.1.1). These adapters can use SEKT technology directly or can use SIP. Although we allow either of these alternatives, the best choice is to integrate in SIP (Technical Integration Layer).

The next table (Table 6.1) summarizes the SEKT technology used by Legal Case Study applications.

We developed all related ontologies according to guidelines of the DILIGENT methodology. Figure 6.3 show integration in SEKT of ontology engineering. The generation of the ontologies (OPJK, QTO and JTO) are described in section 8.

Figure 6.1: SEKT Integration diagram of Expert System
Figure 6.2: SEKT Integration diagram of Jurisprudence System

Figure 6.3: SEKT Integration diagram of ontology engineering in Legal Case Study
Table 6.1: SEKT Technology in Legal Case Study

7 Ontology of Professional Judicial Knowledge.

As we describe on section 2.1.3, to find the best match between the question stored in the repository and the question made by the user, we use an ontology in the final stage to obtain the semantic distance between them. The ontology that we use to make this process possible, is the ontology described at [22], OPJK “Ontology of Professional Judicial Knowledge”. This ontology has been developed by the legal case study team and has been learnt from scratch from the competency questions posed by the judges during their interviews. Modelling this professional judicial knowledge demands the description of this knowledge as it is perceived by the judge and the abandonment of dogmatic legal categorizations.

The Ontology of Professional Judicial Knowledge has been extracted from the selection of relevant terms from nearly 200 competency questions and has, currently, nearly 50 concepts, 100 relations and more than 300 instances. This results from a choice to minimize the concepts at the class level when possible in favour of creating instances and relations.

The ontology used by the prototype system described in this document is modified as a result integrating the PROTON ontology. Figure 7.1: OPJK shows the hierarchy of the OPJK concepts.
7.1 PROTON Integration.

SEKT complementary know-how facilitates the integration of this domain ontology, the OPJK ontology, into PROTON\textsuperscript{13} (PROTo Ontology), developed by Ontotext Lab as a light-weight upper-level ontology, which serves as a modelling basis for a number of tasks in different domains. The PROTON ontology [22] contains about 300 classes and 100 properties, providing coverage of the general concepts necessary for a wide range of tasks, including semantic annotation, indexing, and retrieval. The design principles can be summarized as follows

(i) domain-independence;
(ii) light-weight logical definitions;
(iii) alignment with popular metadata standards;
(iv) good coverage of named entities and concrete domains (i.e. people, organizations, locations, numbers, dates, addresses).

The ontology is encoded in a fragment of OWL Lite, split into four modules: System, Top, Upper, and KM (Knowledge Management) (see Figure 7.2).

\textsuperscript{13} http://proton.semanticweb.org/
It is important to note that the knowledge modelled in OPJK corresponds to the professional (not the purely theoretical) knowledge contained within the judiciary and transmitted only between its members. Professional knowledge encodes a specific type of knowledge related to specific tasks, symbolisms and activities [1] and enables professionals to perform their work with quality [11]. Legal professions, especially the judiciary, not only share among themselves a portion of the legal knowledge constituted by legal language, statutes and previous judgments, but also knowledge related to personal behaviour, practical rules, corporate beliefs, effect reckoning and perspective on similar cases, which remain tacit.

From this point of view, the design of legal ontologies requires not only the representation of normative language but also the representation of the professional knowledge derived from the daily practice at courts, a knowledge that is not being captured by the current trends in legal ontology modelling. Modelling this professional judicial knowledge demands the description of this knowledge as it is perceived by the judge and the attunement of dogmatic legal categorizations; the assumption that their reasoning process follow some specific dogmatic patterns is not required: That is to say, judicial knowledge is not only knowledge about the content of statutes, rules and procedures, but about cases and practical consequences as well. "Attunement" means that the categories and schemes made up by lawyers and legal scholars are usually filtered through this judicial experience.

Capturing this professional knowledge is a time consuming and meticulous process that requires the use of different sociological techniques (field work, etc.) and the gathering of empirical data, which will influence ontological modelling, as learning ontologies from this data “situates” the knowledge. Thus, it is necessary that the methodology followed during the construction of this ontology focuses on the

---

**Figure 7.2: PROTON (PROto ONtology Modules).**
maintenance of this point of view. For this reason, OPJK has been developed following the middle-out strategy [13] that consists of the specification or generalization, when needed, of the identified terms. The integration of these terms into a top ontology has to take into account all this “situated knowledge”.

This integration not only allows the reuse of an existing upper ontology, PROTON, but also allows the maintenance of the necessary “professional” trait of OPJK. This integration has taken place at two stages: OPJK concept into PROTON concept integration, and the reuse of existing PROTON relations.

7.1.1 Class Integration

The first part of the integration process consisted mainly in generalizing OPJK concepts taking into account the System and Top modules of PROTON, and fully incorporating the meta-level primitives contained in the System module (i.e. “Entity”) as the application ontology.

As stated above, the top layer of PROTON is the best level to establish the alignment with OPJK. It has proved to be domain independent, and its easy understanding and usage have been essential to guarantee this integration. The primary classes Abstract, Happening and Object were incorporated straightforwardly, although Abstract needed the introduction of a specific subclass AbstracciónLegal [LegalAbstraction] for organizational purposes. With this domain specific consideration, the OPJK classes CalificaciónJurídica [LegalType], Jurisdicción [Jurisdiction] and Sanción [Sanction] could be better related, and specific relations between them, not shared by the rest of classes/instances within Abstracción [Abstract].

The class of entities Happening, which includes Event, Situation and TimeInterval is able to directly incorporate the fundamental OPJK classes Acto [Act] (ActoJurídico [JudicialAct]), Fase [Phase] and Proceso [Process]. These classes contain the taxonomies and relations related to all legal acts, to the different types of judicial procedures (subdivided within civil and criminal judicial procedures) and the different stages that these procedures can consist of (period of proof, conclusions,...). A necessary reference has to be made to the introduction of the class Role in PROTON, which allowed the distinction of situations where an agent (Organization or Person) might play a part in situations that differ from the more business-related JobPosition [22]. In the case of OPJK, the class Role contains the concepts and instances of procedural roles (RolProcesal) that an agent might play during a judicial procedure.

Finally, Object includes the top OPJK classes Agent and Statement that are generalizations for Document, and Location, necessary concepts to contain, within others, the organizational taxonomy of courts (OrganoJudicial), and judicial documents (Contrato [Contract], Recurso [Appeal], Sentencia [Judgment], etc.).

15 out of 20 of the Top Module classes were taken as super classes of OPJK.

7.1.2 Inherited Relations

The specificity of the legal (professional) domain requires specific relations between concepts (normally domain-related concepts as well). However, most existing
relations between the Top module classes taken from PROTON have been inherited and incorporated. It has not been necessary for the usage of the Iuriservice prototype to inherit all PROTON relations, although most of the relations contained in PROTON had already been identified as relations between OPJK concepts.

The following relations (not a comprehensive list) have been inherited from the existing relations within the Top module concepts: Entity hasLocation, Happening has endTime and startTime, Agent is involvedIn (Happening), Group hasMember, an Organization has parent/childOrganizationOf (Organization) and is establishedIn, and, finally, Statement is statedBy (Agent), validFrom and validUntil.

In conclusion, PROTON ontology contains about 300 classes and 100 properties (400 terms). OPJK has integrated around a 10% of PROTON. The results of this integration were presented in [9].

8 Ontology Generation and the Legal Case Study.

8.1 Description of the Problem

Iuriservice is a web-based application that retrieves answers to questions raised by newly recruited judges in the Spanish Judiciary. This system offers access to a frequently asked questions (FAQ) database through a natural language interface, where the judge describes the problem at hand and the application responds with a list of relevant question-answer pairs that offer solutions to that particular problem. Within this system, ontologies are being used to provide a more accurate search than the basic keyword search [2] and [3].

The primary goal of the task was to develop technology for recognizing semantic concepts from the text, specifically from the questions posed by the judges. It is hoped that such concepts will be more useful for matching questions with the answers than keywords alone. Furthermore, these concepts are grounded: they are detectable from the text automatically. Of course, to some extent the concepts need to be bootstrapped by manually annotating a corpus of questions or judgements. But from this corpus we gain the ability to automatically annotate further questions and judgements.

8.2 Data collection and internal structure

Two main sources of data were used: the corpus of questions, extracted from the interviews with the judges, as presented in D 10.2.1 [5], and the newly built corpus of judgements from the La Ley system.

8.2.1 Collecting questions

The corpus of 756 questions was extracted from data gathered through an ethnographic survey [6]. Then, with the participation and collaboration of the General Council of the Judiciary (GCPJ), the questions containing dilemmas (mostly

---

14 Developed by Intelligent Software Components, S.A. (iSOCO, http://www.isoco.com) and the Institute of Law and Technology (UAB)
regarding best-practices during on-duty hours or staff management), were answered by professionals (the tutors of the Spanish School of the Judiciary).

8.2.2 Collecting judgments

Our La Ley sample contains a sample of over 350,000 judgements. In December 2004, the La Ley system contained 482,000 judgements [5], so our sample represents a considerable portion of the whole database. The database includes valuable copyrighted metadata, especially the summary of each judgement, so it cannot be publicly available.

The judgements were obtained using a custom system for crawling, developed in WP1 by JSI15. It simulates the actions of a user within the Internet Explorer browser. It was not possible to use the focused crawler of D 1.1.1 [19] because the La Ley system involves logging in with the user name and a password, and the session must be maintained after the identification. Moreover, the search system makes heavy use of JavaScript. The reliability of the database is low, especially during the hours of peak access, so numerous errors stalled the process (these errors had to be recovered from). For increased performance, we disabled the download of images. In order not to worsen the experience of other users, crawling was performed primarily during the night, holidays and weekends. In summary, the crawling was far more time-consuming and labour-intensive than was anticipated.

The search system only returns results to a posed query. Since the search system restricts the number of returned results to approximately 300, it was necessary to formulate queries that are exhaustive, but which yield a relatively small number of results. We thereby searched the judgements that have a particular sentence number and year. The sentence number is not unified across courts – instead, each court follows its own numbering sequence. We then perform searches with increasing sentence numbers, for as long as there are judgements.

8.2.3 Structure of judgement.

The retrieved judgements are in the HTML format. The whole judgement has a consistent internal structure, and sections of the judgement are annotated with the HTML DIV and SPAN tags. Some tags used are as follows:

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>juris-nombre-sentencia</td>
<td>number of sentence</td>
</tr>
<tr>
<td>identificador-la-ley-juris</td>
<td>identification of the sentence</td>
</tr>
<tr>
<td>juris-etiqueta-texto</td>
<td>beginning of the text</td>
</tr>
<tr>
<td>juris-encabezamiento</td>
<td>description of the judgement</td>
</tr>
<tr>
<td>juris-encabezamiento-lugar</td>
<td>place of judgement</td>
</tr>
<tr>
<td>juris-encabezamiento-fec</td>
<td>date of the judgement</td>
</tr>
<tr>
<td>juris-encabezamiento-iter</td>
<td>full text of the description</td>
</tr>
<tr>
<td>hechos</td>
<td>history of the case</td>
</tr>
<tr>
<td>hechonumero</td>
<td>individual paragraph of the history</td>
</tr>
<tr>
<td>juris-hechos-probados</td>
<td>findings of fact</td>
</tr>
<tr>
<td>fundamentos</td>
<td>legal grounds for decision</td>
</tr>
<tr>
<td>fundamnumero</td>
<td>individual paragraph of the grounds</td>
</tr>
<tr>
<td>fallo</td>
<td>decision</td>
</tr>
</tbody>
</table>

15 http://www.ijs.si/
The metadata header of the case contains the following META tags:

<table>
<thead>
<tr>
<th>META Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAY_TEXTO</td>
<td>is the text included?</td>
</tr>
<tr>
<td>TRIBUNAL</td>
<td>number of the court</td>
</tr>
<tr>
<td>FECHA</td>
<td>date</td>
</tr>
<tr>
<td>SECCION</td>
<td>number of the section</td>
</tr>
<tr>
<td>TIPO_RES</td>
<td>type of the resolution</td>
</tr>
<tr>
<td>JURISD</td>
<td>number of the jurisdiction</td>
</tr>
<tr>
<td>CABECERA</td>
<td>summary</td>
</tr>
<tr>
<td>PONENTE</td>
<td>author of opinion</td>
</tr>
<tr>
<td>TITULO</td>
<td>title of the judgement</td>
</tr>
<tr>
<td>NUM_SENTENCIA</td>
<td>number of the judgement</td>
</tr>
<tr>
<td>ANNO_SENTENCIA</td>
<td>year of the judgement</td>
</tr>
<tr>
<td>NUM_RRP</td>
<td>number of the appeal</td>
</tr>
<tr>
<td>ANNO_RRP</td>
<td>year of the appeal</td>
</tr>
</tbody>
</table>

8.3 Hand-Made Semantic Structure

8.3.1 Pre-existing semantic annotations of questions

Initially, the team of domain experts identified several sub-domains (topics) intuitively gathered from the questions. The detection of sub-domains led, at the beginning, to a classification of questions into different sub-domains to ease the construction of the ontology, which focused, at first, on two specific sub-domains: on-duty and family issues (containing gender violence and protection/injunction orders). The manual classification of questions into sub-domains resulted in the list shown below, and in Figure 8.1:

1. On-duty
2. Family issues
2.1. Gender violence
2.2. Minors
2.3. Divorce/separation/custody
2.4. Other
3. Immigration
4. Property
5. Sentencing
5.1. Execution
5.2. Breach of order/judgment
5.3. Form
5.4. Notification
6. Proceedings
6.1. Evidences
6.2. Competence
6.3. Criteria
7. Court office
7.1. Organization
7.2. Civil servants
7.3. Prosecution
7.4. Infrastructure
7.5. Security
7.6. Informatics
7.7. Experts
7.8. Police and security
7.9. Incompatibilities
7.10. Relations with the GCJP
7.11. Other
7.12. Lawyers
8. Business Law
9. Traffic accidents
10. Criminal Law
10.1. Drugs
10.2. Organized crime
10.3. Theft/Robbery/Burglary
8.3.2 Pre-existing semantic annotations of judgements

The hand-made summary of a judgement can be seen as an extract of semantically relevant information of the judgement. In contrast to the remainder of the judgement, the summary serves as an extract of the ‘general’ information in the judgement, as distinguished from the ‘details’ in the judgement. Very many different authors wrote the judgements, there are nuances in the way they are written. Finally, not all the judgements are in Spanish. For that reason, the summaries tend to be far more unified in their style and content.

The summary is a selection of phrases. Some of the phrases are capitalized, especially the first part of the summary:

RESPONSABILIDAD EXTRACONTRACTUAL. ACCIDENTES DE CIRCULACIÓN. Furgoneta indebidamente frenada que causa daños en el vehículo del demandante. Prueba testifical que acredita la versión de los hechos dada por el actor. Procedencia de la indemnización reclamada.

In the above example of a summary, “responsabilidad extracontractual” is the primary topic of the judgement. The “accidentes de circulación” is a secondary topic, which refines the primary one. It is not, however, standard whether the secondary topic is capitalized or not.

ASESINATO INTENTADO. Animus necandi. Golpes repetidos con un objeto contundente en el cráneo de la anciana. ALEVOSÍA. Eliminación de las posibilidades de defensa de la víctima. PARENTESCO. Agravante. Ataque del nieto a su propia abuela. HOMICIDIO INTENTADO. Golpes en la cabeza de una segunda víctima. ROBO CON VIOLENCIA. Uso de instrumento peligroso. REPARACIÓN DEL DAÑO. Atenuante. INTOXICACIÓN ETÍLICA. Eximiente incompleta.

In the above example, a number of phrases are capitalized, and several of them are indeed topics (homicidio – homicide, robo con violencia – violent robbery) that appear elsewhere as primary. It is not clear whether the role of these secondary
capitalizations are to refine a primary topic or to provide a list of synonyms or related topics. For that reason, we only extract the primary topic. Nevertheless, such capitalizations may assist in the construction of an ontology through specifying facets, properties or attributes of a certain case.

The extraction of the primary topic also accounts for some errors, such as the absence of the full stop after the initial capitalization, or a few non-capitalized words inside the topic. If no capitalized topic is given at the front of the summary, the extraction procedure takes the first sentence even if it is not capitalized. In general, the handling for several special cases was developed to cope with some frequent non-standard cases.

Generally, a number of summaries should belong to the same topic for that topic to be well-defined. For that reason, we can only afford to extract frequently occurring topics. Topics that occur infrequently will not have enough supporting text, and as a result their semantics is not well-defined. To begin with, we assumed that each unique string of letters in the place of the primary topic marked a separate topic. There are, however, two reasons why this may not be true. First, there are numerous misspellings of frequent topics. This list contains a part of the possible spellings of the topic *responsabilidad extracontractual*, along with the number of cases present in the database:
Only the frequent topics (occurring in more than 4-5 judgements) are considered. The second problem is that there are also different phrases that have a similar meaning. For example, several different phrases all carry a similar meaning of “homicide”:

HOMICIDIO 890
HOMICIDIOS 2
DELITO DE HOMICIDIO 1
DELITO_INTENTADO_DE_HOMICIDIO 1
DEL_HOMICIDIO 1

The solution to this problem can be achieved by applying ontology generation, so that even different phrases can be joined into the same class of a judgement. Some of the primary topics are distinctly broad and label a tremendous number of judgements:
For example, there have been 4900 judgements with the primary topic of urbanism. In this case, the primary topic is probably insufficient as a descriptor of the topic, and additional subtopics could be used to better specify the semantically interesting aspects of the judgements. Again, ontology generation may help discover the structure inherent to these 4900 judgements.

Another problem is that the primary topics partially include the facets of an individual topic, and co-appearances of two topics in a single judgement. For example, consider this sample of subtopics of homicide:

- HOMICIDIO_INTENTADO 72
- HOMICIDIO_INTENTADO_Y_AMENAZAS 1
- HOMICIDIO_Y_ASESINATO 1
- ATENTADO_Y_HOMICIDIO 1
- CONDUCCION_TEMERAIRA_Y_HOMICIDIO_IMPRUDENTE 1
- DELITOS_CONTRA_LA_SEGURIDAD_EN_EL_TRABAJO_Y_HOMICIDIO_IMPRUDENTE 1
- DELITO_DE_HOMICIDIO_EN_GRADO_DE_TENTATIVA 1
- DELITO_DE_HOMICIDIO_IMPRUDENTE 1
- DELITO_DE_HOMICIDIO_POR_IMPRUDENCIA_GRAVE 1
- DELITO_DE_HOMICIDIO_Y_SUS_FOMAS 1
- HOMICIDIO_CONSUMADO 1
- HOMICIDIO_CULPOSO 1
- HOMICIDIO_DOLOSO 1
- HOMICIDIO_EN_GRADO_DE_TENTATIVA 31
- HOMICIDIO_EN_GRADO_DE_TENTATIVA_EN_CONCURSO_IDEAL_CON_HOMICIDIO_IMPRUDENTE 1
- HOMICIDIO_FRUSTRADO 2
- HOMICIDIO_IMPRUDENTE 120
- HOMICIDIO_IMPRUDENTE_Y_LESIONES 1
- HOMICIDIO_IMPRUDENTE_Y_OMISION_DEL_DEBER_DE_SOCORRO 1
- HOMICIDIO_POR_IMPRUDIENCIA 26
- HOMICIDIO_POR_IMPRUDIENCIA_CIRCULATORIA 1
- HOMICIDIO_POR_IMPRUDIENCIA_GRAVE 31
- HOMICIDIO_POR_IMPRUDIENCIA_LEVE 2
- HOMICIDIO_POR_IMPRUDIENCIA_MEDICA_GRAVE 1
- HOMICIDIO_POR_IMPRUDIENCIA_PROFESIONAL 2
- HOMICIDIO_POR_IMPRUDIENCIA_TEMERAIRA 3
- HOMICIDIO_Y_LESIONES 4
- HOMICIDIO_Y_LESIONES_IMPRUDENTES 1
- HOMICIDIO_Y_LESIONES_POR_IMPRUDENCIA 1
- HOMICIDIO_Y_LESIONES_POR_IMPRUDIENCIA_GRAVE 3
- DELITO_DE_HOMICIDIO_FRUSTRADO_EN_CONCURRENCIA_CON_EL_DELITO_DE_LESIONES 1
- HOMICIDIO_Y_MALOS_TRATOS 1
Clearly, there are even more synonymous phrases: the rarer the topic, the lower the probability that the semantic annotator would know the standard phrase to name it. Another important observation is that the name of the topic already indicates the content: we do not need to learn the meaning, as the meaning is already contained in the name.

In summary, the topics are visible from the summaries. However, they are not organised as well as they could be. Specifically, the following inadequacies can be solved by building and using a topic ontology:

- The organization of related topics into a taxonomy.
- A controlled, standardized vocabulary of topics.
- The refinement of a frequent topic through specification of details, either through subtopics or through the attributes of a topic.
- Quality control of the semantic annotation of judgements.

### 8.4 Application of natural language technologies: lemmatization

Most of the text is in Spanish. Surprisingly, some of the judgements are also in Catalan. It has been found that stemming does not give satisfactory results [23]. For that reason, a freely available tool FreeLing 1.2 was used [4]. The FreeLing tool performs tokenization, sentence splitting, morphological analysis, NE detection, date/number/currency recognition, PoS tagging, and chart-based shallow parsing.

FreeLing is used as a preprocessing tool. For example, this is the analysis performed by FreeLing on the sentence:

> “En el segundo motivo se denuncia aplicación indebida de normas sustantivas alegándose en primer lugar infracción del art. 24 de la Constitución Española”:

<table>
<thead>
<tr>
<th>En</th>
<th>En</th>
<th>SPS00</th>
</tr>
</thead>
<tbody>
<tr>
<td>El</td>
<td>El</td>
<td>DA0MS0</td>
</tr>
<tr>
<td>Segundo</td>
<td>segundo</td>
<td>AO0MS0</td>
</tr>
<tr>
<td>Motivo</td>
<td>motivo</td>
<td>NCMS000</td>
</tr>
<tr>
<td>Se</td>
<td>él</td>
<td>P0300000</td>
</tr>
<tr>
<td>Denuncia</td>
<td>denunciar</td>
<td>VMIP380</td>
</tr>
<tr>
<td>Aplicación</td>
<td>aplicación</td>
<td>NCFS000</td>
</tr>
<tr>
<td>Indebida</td>
<td>indebida</td>
<td>AQ0FS0</td>
</tr>
<tr>
<td>De</td>
<td>de</td>
<td>SPS00</td>
</tr>
<tr>
<td>Normas</td>
<td>norma</td>
<td>NCFP000</td>
</tr>
<tr>
<td>Sustantivas</td>
<td>sustantivas</td>
<td>NCFP000</td>
</tr>
<tr>
<td>Alegándose</td>
<td>alegar</td>
<td>VMG0000</td>
</tr>
<tr>
<td>En</td>
<td>en</td>
<td>SPS00</td>
</tr>
<tr>
<td>Primer</td>
<td>primero</td>
<td>AO0MS0</td>
</tr>
<tr>
<td>Lugar</td>
<td>lugar</td>
<td>NCMS000</td>
</tr>
<tr>
<td>Infracción</td>
<td>infracción</td>
<td>NCFS000</td>
</tr>
<tr>
<td>Del</td>
<td>del</td>
<td>SPCMS</td>
</tr>
<tr>
<td>art.</td>
<td>art.</td>
<td>NC00000</td>
</tr>
<tr>
<td>24</td>
<td>24</td>
<td>Z</td>
</tr>
<tr>
<td>De</td>
<td>de</td>
<td>SPS00</td>
</tr>
<tr>
<td>La</td>
<td>el</td>
<td>DA0FS0</td>
</tr>
<tr>
<td>Constitución Española</td>
<td>Constitución Española</td>
<td>NP00000</td>
</tr>
</tbody>
</table>
We can observe named entity extraction (Constitución Española), lemmatization (the second column), and part-of-speech tagging (third column).

8.5 Application of semantic web technologies: topic ontologies in RDF

The topic is a semantic annotation of a document that people instinctively use to search and organize information. While we can semantically annotate many facets of a document, the notion of topic covers the document as a whole: it is the conceptualization of the document with the greatest breadth. When we seek information, our primary intention is usually to discover information on the same topic as is the topic of our problem. Topic can be distinguished from other descriptors of a document, such as language, style, punctuation, grammar, reputation, graphical design, author, date, and so on.

Formally, a topic can be viewed simply as a class or a concept. In this setting we can say that an instance of the class “EU Constitution” is a document whose topic is “EU Constitution”. Thus, a document is a kind of an instance or an individual, and a topic is a kind of a concept. So, the ontological notions and structures are fully applicable to describing documents and their topics. Technically, however, this conceptual structure can be mapped to the RDF structure in many ways. The one used in PROTON assumes that both topics and documents are instances. Moreover, the transitive hasSubject relation associates a document with its topics. Such a restricted family of ontologies with a somewhat different semantics are referred to as topic ontologies [14] to distinguish them from more general schema ontologies [22].

We will now describe how the topics and documents are represented in the RDF format used by OntoGen [12]. We will treat both topics and documents to be instances. To associate them, we will use the transitive hasSubject predicate to associate a document with its topics:

There is a very large number of different topics. For that reason, it is often convenient or necessary to organize them. The dominant structure is indeed the transitive hierarchical structure of a taxonomy. For example, if Marketing is a subtopic of Business, then every document that is about Marketing will also be about Business.
To distinguish documents about Business in general a subtopic of GeneralBusiness needs to be created.

The structuring of the topics serves two main purposes. The first is to allow the user to carry out a quick search for a specific topic by following the tree of topics from the root to a particular leaf topic. The second is to establish semantic connections between similar topics. Some topics can be considered similar through our background knowledge (similarity of authors, similarity of ideas), and it is this knowledge that helps fill in the gaps, even if the data itself is insufficient.

Multiple topics can be organized into a hierarchical topic ontology or a taxonomy. In general, a document may belong to multiple topics. For example, a newspaper article about a bomb explosion in Iraq can belong both to the topic of US Foreign policy, to the topic of Middle East, and to the topic of terrorism. Sometimes, one of the topics is primary, the other topics being used to refine it. This is apparent from the ordering of topic descriptions, where we list the topics from the most important ones downwards towards marginally important ones.

8.6 Application of machine learning technologies: topic detection

The fundamental idea of text categorization is to automatically determine the category (topic) of a piece of text (such as a document, or a natural language question). There are many applications for such technology. For example, if we can detect semantic concepts in text, we can automatically create a semantic description from the text. By detecting the same concept both in the question and in the answer, we can find that particular answer to be relevant. By being able to detect the primary topic of a judgement, we can automatically create a part of the summary. We can perform “quality assurance” of the summaries. We can even detect synonymy and other relations between topics using text categorization. If, for example, “homicide” is predicted with high confidence for all summaries tagged with “homicides”, we can suspect that the two phrases are synonymous, for example.

The enormous database of judgements can be used as a leverage for the purpose of detecting concepts in questions and answers. For example, if we learn to identify the topic of “homicide” in the judgements, we can detect the same topic also in an answer. Or, the analysis of words and phrases in the interviews with the judges also applies to the database of judgements. In all, the question answering task in the case study can benefit from the database of full-text judgements.

Classification technology, an ubiquitous technique in machine learning, is required. A classifier (or a classification model) detects topics from documents automatically. Since the classifier cannot work directly with the full text, the documents are represented with “bag of words” (as is usual in text categorization [21]). Each word (or more generally speaking feature) is represented with a number proportional to the frequency of its appearance in the text. If the feature was not detected in the text, this number will be zero.

In the case of Spanish text, the features correspond to lemmas of the words: the tense, gender or other grammatical properties of a word are ignored. In some cases, features can denote multi-word expressions with a specific meaning (such as “orden de
D10.3.1 /Prototype

protección”). As usual in text categorization, we have performed some pre-processing of the text to remove punctuation and stop-words, including some prepositions.

The classifier uses the support vector machine (SVM) approach, with a dot product kernel ([24],[13]). This is perhaps the most frequently used technique for the classification of text, and is considered one of the best [18].

Classification is a supervised technique, meaning that we need the topic ontology populated with documents/text in order to construct a classifier. One way of using the classifier is to annotate previously unseen documents in order to evaluate the ontology against the documents - ontology grounding [15]. If a topic turns out not to be grounded, there are several ways of remediying the problem:

- Increase the amount of training data.
- Redesign or simplify the topic ontology.
- Increase the amount of background knowledge:
  - Addition of features that recognize phrases (such as “civil law” as a single feature compared to “civil” and “law”)
  - Specification of feature importance (some features, such as the legal terms, are considered more important than other features, such as arbitrary adjectives)
  - Removal of features that are known to be irrelevant or that cannot be used by the classifier (such as punctuation and determiners)

It is usually insufficient to merely analyze the classification performance of a model. To diagnose the reasons and courses of remedy, we need to examine the problems in more detail. Specifically, we need to understand the causes of the errors. Such causes will provide insight about the missing background knowledge, which can be introduced manually. To facilitate the analysis, we employ two techniques:

- Identifying the decisive features for each concept.
- Identifying the instances where the concept is not identified correctly.

The SVM classifier encodes each feature with a real-valued variable. All the features are encoded in a feature vector $x$. The topic is encoded with the variable $y$ that takes the value of +1 when the topic is present and -1 when it is absent. The classifier then acts as a function $y = f(x)$. SVM with the dot product kernel is associated with a function that takes the following form:

$$f(\vec{x}) = \begin{cases} 
-1; & \sum_i \alpha_i y_i (\vec{x}_i \cdot \vec{x}) < 0 \\
+1; & \text{else}.
\end{cases}$$

Here, $y,\vec{x}_i$ are the labelled training instances, and $\alpha_i$ are the parameters of the model. While this sum is over the training instances with nonzero $\alpha_i$, an alternative is to convert it into a sum over the features [16] In that case, $f$ takes such a form:

$$f(\vec{x}) = \begin{cases} 
-1; & d + \sum_j w_j [\vec{x}]_j < 0 \\
+1; & \text{else}.
\end{cases}$$
Here, \([\tilde{x}]_r\) denotes the \(j\)-th dimension of vector \(\tilde{x}\). We can interpret the weights \(w\) as some sort of an importance of each individual feature. However, rare words would appear to have a very high importance, while in fact they only apply to rare cases. Instead, our measure of feature importance for feature \(j\) is going to be the net leverage \(q\) across all the training instances:

\[
q(j) = \sum_i w_j [\tilde{x}_i]_j
\]

Positive sign of net leverage will indicate that the particular feature votes in favor of a topic, and a negative sign will indicate votes against the topic.

While the net leverage allows the overall diagnosis of the corpus, leverage can also be computed only for a particular document. The document may sometimes be misclassified: a document having the topic but classified as not having the topic is a false negative, while a document not having the topic but classified as having the topic is a false positive. By examining misclassifications we can see what went wrong, and seek courses of remedy.

### 8.7 Using topic detection to evaluate a manually-constructed topic ontology

During the manual conceptualization of Ontology of Professional Judicial Knowledge (OPJK), the domain experts and ontology engineers in the team realised that most of the questions regarding on-duty and family issues were necessarily related to procedural moments (phases or acts, proceedings). Moreover, decision-making and judgements were also necessarily related to proceedings and domestic violence (included within family issues) was necessarily related to Criminal Law, thus related to the other sub-domains being modelled. This fact showed, again intuitively, that the questions could have been misclassified and also that the questions could refer to more than just one intuitive sub-domain at the same time.

This intuitive conclusion was confirmed by the results provided by text categorization described in section 8.6. In simple terms, the classifier attempted to learn by which keywords to distinguish the topics or sub-domains of questions. Ideally, the classifier would automatically, correctly and unambiguously decide what sub-domain a particular question belongs to. In practice, however, the classifier will make mistakes. Some of the mistakes might indicate that the categories are not specified with a sufficient amount of detail. Other mistakes may indicate incorrectly labelled questions, but there might also be questions that can be correctly classified in several ways.

The classifier first analysed the intuitive classification in sub-domains and provided us with two different statistical reports. The first report was a list of keywords for each sub-domain. The list of keywords showed which terms the classifier thought to be in favour or against of that sub-domain. The second report was a diagnosis for each of the intuitively and manually classified questions, listing arguments for or against that classification.

#### 8.7.1 Analysis of keywords
To display the “keywords” of a particular sub-domain, we simply build the classifier to detect the sub-domain. Then, we seek out those features that have the highest net leverage: these features are the ‘arguments in favor”. The “arguments against” are the features with the lowest negative net leverage.

Analysing the lists of keywords for each sub-domain, the domain experts detected that most terms stated to be in favour had no relevant meaning for that particular sub-domain, some terms stated to be against a sub-domain could actually be relevant for the sub-domain and some terms neither in favour nor against had no more special relation to that particular sub-domain than to another.

For example, if we take the example of the on-duty sub-domain (shown above), legal domain experts can identify that on-duty [guardia], corpse [cadaver], doctor [medico], corpse removing [levantamiento], entry [entrada], urgent [urgence], police [policia] and search [registro] are relevant terms for the on-duty period. However, suit [asunto], transfer [traslado], person [persona], court order [auto], social [social] and to refer [remitir] had no relevant meaning for that particular sub-domain.

Moreover, the list of keywords against a particular question being classified into the on-duty sub-domain contains some terms which can be identified by domain experts as being in favour of a question belonging to the on-duty sub-domain. The terms autopsy [autopsia] and telephone [telefono] are especially relevant to on-duty questions as most autopsies and line tappings are requested during on-duty hours. We also have to bear in mind that most on-duty requests are made through a phone call to the judge (who might be home).

Also, some other terms of the list against questions being classified into the on-duty sub-domain could be identified as relevant to the on-duty questions but could be relevant to other sub-domains as well. Case [caso], lawyer [abogado], minor [menor], trial [juicio], civil servant [funcionario] and offence [delito] are relevant keywords to
the on-duty sub-domain but also to the family issues sub-domain (minor), to the
criminal law sub-domain (offence), to the court office [oficina judicial] sub-domain
(civil servant and lawyer), to the proceedings sub-domain (trial) or to any subdomain
(case).

A particular feature may be an argument against a sub-domain even if it does appear
in the corresponding questions. This can be explained by the fact that the particular
feature appears even more frequently in other sub-domains. But as some sub-domains
only have a small number of examples (questions), the classifier can make mistakes..

8.7.2 Analysis of misclassifications

Because all the questions have been manually classified by the experts, we know what
the true classification of a question is. If the question is then shown to the classifier,
we can check if the automatic classification is consistent with the manual one. If there
has been a mistake, we can examine what features were decisive in the classification.
We show five features with the highest and five features with the lowest leverage for
the wrong classification. Furthermore, we do the same for the correct classification.

The errors may arise for four reasons: overvaluing of arguments in favor of wrong
decision and against the correct decision, and undervaluing of arguments in favor of
the right decision and arguments against the wrong decision. In practice, the classifier
might not detect an important feature, or might detect an irrelevant feature.

When this analysis was related to the analysis made by the same domain experts of
the results of the diagnosis for each of the questions originally classified in the on-
duty sub-domain, they could detect that:

1. Some on-duty questions had been misclassified in that sub-domain and
   they clearly belonged to another (see list below).  

| ¿Qué hago con el procedimiento cuando mando análisis de ADN y tardan casi un año en llegar los resultados? |
|-------------------------------------------------|-------------------------------------------------|
| True=guardia Predicted=proceso/pruebas           |                                                 |
| * Keywords in favor of guardia                   | * Keywords against guardia                       |
| 0.00 ( 0) estar                                  | 1.35 ( 4) mandar                                 |
| 0.00 ( 0) juzgar                                 | 0.36 ( 6) llegar                                 |
| 0.00 ( 0) único                                  | 0.06 ( 6) ano                                    |
| 0.00 ( 0) guardia                               | 0.04 ( 1) ?                                     |
| 0.00 ( 0) permanente                            | 0.00 ( 4) procedimiento                         |

| * Keywords in favor of proceso/pruebas           | * Keywords against proceso/pruebas               |
| 2.99 ( 6) analisis                              | 0.32 ( 4) procedimiento                         |
| 1.85 ( 7) casi                                  | 0.04 ( 1) ?                                     |
| 1.85 ( 7) adn                                   | 0.00 ( 0) estar                                 |
| 1.29 ( 6) ano                                   | 0.00 ( 0) juzgar                                |
| 1.11 ( 6) resultado                             | 0.00 ( 0) único                                 |

2. Some on-duty questions were correctly classified into the on-duty period
   but were also relevant to other sub-domains (see list below).
¿Cómo proceder en aquellos casos en que una mujer interpone denuncia por malos tratos, pero pide expresamente que no se acorde ninguna medida de alejamiento?

True=guardia Predicted=familia/violencia_domestica

<table>
<thead>
<tr>
<th>* Keywords in favor of guardia</th>
<th>* Keywords against guardia</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.63 ( 6) ninguno</td>
<td>1.55 ( 4) trato</td>
</tr>
<tr>
<td>0.88 ( 3) pedir</td>
<td>1.47 ( 4) malo</td>
</tr>
<tr>
<td>0.47 ( 4) proceder</td>
<td>0.75 ( 4) pero</td>
</tr>
<tr>
<td>0.23 ( 2) .</td>
<td>0.65 ( 3) caso</td>
</tr>
<tr>
<td>0.11 ( 4) mujer</td>
<td>0.65 ( 5) alejamiento</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>* Keywords in favor of familia</th>
<th>* Keywords against familia</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.24 ( 4) medida</td>
<td>1.18 ( 4) proceder</td>
</tr>
<tr>
<td>2.85 ( 4) malo</td>
<td>1.06 ( 6) ninguno</td>
</tr>
<tr>
<td>2.71 ( 4) trato</td>
<td>1.00 ( 3) pedir</td>
</tr>
<tr>
<td>2.66 ( 5) alejamiento</td>
<td>0.63 ( 5) aquel</td>
</tr>
<tr>
<td>1.79 ( 4) denunciar</td>
<td>0.40 ( 3) caso</td>
</tr>
</tbody>
</table>

3. Some true on-duty questions were shown wrongly to belong to another sub-domain (see list below).

Me llaman desde el depósito municipal de cadáveres diciendo que ha venido la funeraria pidiendo la licencia para llevarse un cadáver. ¿Qué tengo que comprobar antes de darles la licencia? ¿Qué hay que hacer?

True=guardia Predicted=proceso/pruebas

<table>
<thead>
<tr>
<th>* Keywords in favor of guardia</th>
<th>* Keywords against guardia</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.60 ( 9) cadaver</td>
<td>5.10 ( 13) licencia</td>
</tr>
<tr>
<td>2.25 ( 4) llamar</td>
<td>2.89 ( 6) municipal</td>
</tr>
<tr>
<td>0.88 ( 3) pedir</td>
<td>1.99 ( 5) depósito</td>
</tr>
<tr>
<td>0.00 ( 0) estar</td>
<td>1.27 ( 7) funeraria</td>
</tr>
<tr>
<td>0.00 ( 0) juzgar</td>
<td>1.16 ( 5) antes_de</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>* Keywords in favor of proceso</th>
<th>* Keywords against proceso</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.87 ( 13) licencia</td>
<td>1.63 ( 9) cadaver</td>
</tr>
<tr>
<td>0.79 ( 5) antes_de</td>
<td>0.78 ( 4) dar</td>
</tr>
<tr>
<td>0.60 ( 3) pedir</td>
<td>0.58 ( 2) .</td>
</tr>
<tr>
<td>0.59 ( 6) municipal</td>
<td>0.33 ( 4) venir</td>
</tr>
<tr>
<td>0.35 ( 5) depósito</td>
<td>0.28 ( 4) llamar</td>
</tr>
</tbody>
</table>

In conclusion, the data showed that the classifier encountered several problems in classifying the questions, mostly due to the inefficient original classification. One of the reasons was automatic misclassification, as shown above. However, there was another reason: there was the need to consider the on-duty sub-domain differently. The on-duty sub-domain does not represent a conceptual domain of specific knowledge, it represents a time domain. The on-duty period involves all those questions that happen on a certain time. Therefore, the questions belonging to the on-duty period could and should belong also to the other sub-domains that represent conceptual domains and contain the questions in a certain knowledge area.

Secondly, the classifier misunderstood the relevance of some of the keywords used for argumentation. As stated above, some terms were not relevant for a particular sub-domain. However, the main difficulties were:

1. The usage of completely irrelevant terms for the legal domain (at least on their own) like but [pero], to order [mandar], to follow [seguir], to put [poner], that [ese], work [trabajo]);
2. The inability to capture relevant compound forms. For example: bad [malo] and treatment [trato] are regarded separately, while they are actually a
compound form: bad/ill treatment (malos tratos) is a relevant key-phrase, used to identify the questions related to domestic violence.

Therefore, a conceptual change was adopted and on-duty was regarded, from that point, as a time domain, rather than a sub-domain in itself. Also the data had to be prepared as to reduce the number of errors.

8.8 Recommendations for text preparation

One of the main principles in Corpus Linguistics is that the analyst has to do his best to avoid as much “noise” in the corpus as he can. There are two important pre-processing operations: to unify all the words with a very similar meaning, and to remove the words that do not carry relevant meaning. The first operation is achieved using stemming, lemmatisation and with thesauri (including WordNet). In stemming, we remove the final few letters of the word that carry grammatical nuances but do not distinguish the meaning. For example, we represent words dog, dogs, doggy with dog+.

A verb such as be may take several forms: is, are, were, being, be. During lemmatisation, we replace all these forms that may reflect tense, gender, inflection, irregularity, etc., with the simple canonical one: be. While this particular example may seem contrived, other languages, such as Spanish, have many more such examples. Such languages benefit much more from lemmatisation, while English does well already just with stemming.

The second important practice is removing the words that are deemed not to carry any relevant meaning for the task of determining sub-domains and topics. Such words (or lemmas) are referred to as stop words. There may be two reasons why they are to be ignored. Some words may carry no conceptual meaning on their own (e.g. prepositions, conjunctions, etc). Although a more sophisticated method would be able to gain information from such words, the “bag of words” approach cannot, so such words are mere noise. The second type of words may have meaning on their own, but they do not add any (semantic) value for the purpose of our analysis (e.g. auxiliary verbs). A list of such words is then used to prepare the data before classification or ontology construction.

8.8.1 Irrelevant Grammatical Categories

In any linguistic corpus we may find many “grammatical” or function words that are unlikely to be useful for search. For example, common function words in a text sample in Spanish are de, la, que, el, en, y, a, los, del, etc. These words do have important functions in Spanish, but they hardly contribute any useful information for search or topic classification.

The stop words mentioned above (within others) correspond to different grammatical categories and thereby they are to be tagged in diverse ways by the morphological
analyser we are using. For instance, *de* [of], *en* [in] and *a* [to] are prepositions [SPS00]; *la* [the, fem.], *el* [the, masc.] are a defined, feminine/masculine, singular articles [TDFS0, TDMS0], *que* [that, which] and *y* [and] are a subordinate [CS00] and a coordinate [CC00] conjunction, and so forth.

The words in our corpora tagged with the following standard codes (corresponding to function words) will be considered stop words. The first two letters of the code are sufficient for their identification, according to EAGLES tags (e.g. TD—, CC—, etc.):

- **Articles, Determiners**: Demonstrative (DD—), Possessive (DP—), Interrogative (DT—), Exclamatory (DE—), Indefinite (DI—)
- **Pronouns**: Personal (PP—), Demonstrative (PD—), Possessive (PX—), Indefinite (PI—), Interrogative (PT—), Relative (PR—)
- **Conjunctions**: Coordinated (CC00), Subordinated (CS00)
- **Interjections**: (I)
- **Prepositions**: Adpositions (SPC-), Prepositions (SPS00)
- **Numerals**: Cardinal (MC—), Ordinal (MO—)
- **Punctuation signs**: (F)

The main category of meaningful words that should be included into the stop word list in order to ease the linguistic treatment are the auxiliary verbs. Some of its forms are in fact removed during the lemmatization process, since all the forms of a verb (including compound ones as *ha hecho* [has done]) are reduced to its infinitive form, so that the auxiliary form *ha* [from *haber* “to have”] is deleted as it functions as an auxiliary verb. Since some errors can be made in this process and some auxiliary verbs can be interpreted by the morphological analyser as full verbs, we propose the elimination *a priori* of all auxiliary verbs. The first two letters of its EAGLES code are VA-----.

Another question regarding the classifier is that some verbs like *ser* and *estar* [to be], *tener* [to have something], *haber* [to have, to exist], are used too frequently to have an important value in classification. In other words, neither of these verbs can be seen as defining any special kind of domain, since they may be found in any kind of standard sentence. Our proposal would be to eliminate them also, as we do with the auxiliary verbs. This should be done after the lemmatization [23]—i.e. once the program has reduced all the verbal forms to infinitive—in order to make it easier for the program to identify just the verbal forms *ser, estar, tener* and *haber*.

**8.8.2 Words to be preserved in the corpus**

In general terms, the words that contribute most to the meaning of a text are nouns and verbs. With the exceptions mentioned above, these kind of words have to be preserved in the corpora in order to perform a good search. The codes are:

1. For verbs (main verbs): VM-----
2. For nouns:

---

16 [http://garraf.epsevg.upc.es/freeling/]
In reference to the other categories not mentioned so far (adjectives, adverbs and abbreviations), we think they should be preserved for similar reasons. First, adjectives (AQ----) and substantives (NC— and NP—) —in a broad semantic sense— are considered to form a joint category that some authors call ‘noun’, which means that the adjective has not always been treated as a completely independent category, for it works as a complement for the sense of the substantive. To delete it would imply a great loss of information.

A typical example of this would be *malos tratos* [ill-treatment], which is formed by an adjective *[malos]* plus a substantive *[tratos]*. Secondly, however its strength and importance in sense-adding value is not the adjective’s (at least in terms of occurrence), adverbs (RG—) are still important for our purposes, because they may add important information about the use (thus the sense) of a verb in many cases. An example of this may be *tramitar urgentemente* [to proceed urgently]. Thirdly, abbreviations (Y) (like *etc.*, but also like *LeCrim* [abbreviated form for referring to the Criminal Code]) are very important for our purposes, since the legal discourse makes a great use of them and can be of great value for classification.

### 8.8.3 Domain dependent words

The Ontology of Professional Judicial Knowledge (OPJK), introduced above, currently covers the sub-domains of *guardia* [on-duty period], domestic violence and procedural law. Currently, it has to be completed with issues from criminal and commercial law. Obviously the words —whether they are concepts or instances— which appear in this ontology ought to be especially taken into account in the classification process. In other words, OPJK ought to be seen (for this purpose) as a statement of relevant linguistic terms that are important for capturing the semantics we care about. This ontology is being built from textual protocols extracted from an extended field work, which *per force* implies a strong relation between words and real data. Moreover, the extraction of the words (concepts and instances) has been made manually, mostly extracted by hand by the legal expert team. Thus the words which appear in OPJK ought to be given some special consideration.

Moreover, as a good number of the terms in OPJK do not correspond to a single word but to a compound form, these compound forms ought to be taken into account as single occurrences and not as two. Typical forms extracted from OPJK are:

- Noun + Preposition + Noun (*juzgado de guardia, diligencia de embargo*, etc.)
- Adjective + Noun (*malos tratos, lesa humanidad*, etc.)
- Noun + Adjective (*documento procesal, enjuiciamiento civil*, etc.)

### 8.9 Application of knowledge discovery: semi-automated topic ontologies

As described above, there were several recommendations derived from the analysis of the manually constructed ontology for classification of questions. Thus, the ontology
could be revised taking into account the recommendations. We have used OntoGen developed in WP1 (D1.7.1) to construct a new topic ontology of the questions.

Two topic ontologies were created using OntoGen, one for the questions corpus and one for the judgements corpus. For both corpora, we performed lemmatization and removed the words from the stop word list as described in the previous section (but not all the recommendations were implemented). For the judgements corpus, a document was actually a concatenation of all the summaries that correspond to a particular primary topic. This operation was needed to allow quick and responsive operation of OntoGen on the usual PCs: working with 350000 judgements would be both time-consuming.

8.9.1 **OntoGen - a tool for semi-automatic ontology construction**

OntoGen [12] applies several techniques from information retrieval and data mining to empower a topic ontology designer. Specifically, one starts with a large set of documents. These documents are to be partitioned into \( k \) categories. The fundamental property of a category is that the documents within it are similar to one another with respect to the features used. Methodologically, this is achieved by latent semantic indexing and clustering. Each of these categories is a subtopic of the root topic that contains all documents. Also, each of these categories can be further categorized into sub-categories. Keyword extraction is used to describe the meaning of a category to the designer in the form of a small set of characteristic keywords, who may subsequently choose a different name.

8.9.2 **Question Topic Ontology.**

Question Topic Ontology (QTO) was constructed using OntoGen taking into account both the first intuitive classification and the manual classification. These classifications were based on legal expertise and, thus, the topics identified were relevant to the construction of this ontology. The previously detected topics were maintained if they could be detected from the suggested list of topics, defined by a list of keywords, provided by OntoGen on the corpus of questions. Moreover, the topic *guardia* was regarded as the center of the QTO as most of the questions posed by the judges referred to that time, rather than regarded as a topic in itself (at the same level of topics such as *gender violence*, etc.).

Nevertheless, we have to bear in mind that the current version of OntoGen is not able to classify the questions into more than one topic. For that reason, the domain experts had to check carefully, the classification of each question, trying to decide which of the possible topics where it could be included was the best.

For that reason, the classification within topics might not be completely accurate (we have stated that some questions could be classified in more than one topic). However, we will use this preliminary work on question classification using OntoGen to continue analysing the relevance of certain keywords for classification into topics and other functionalities to ensure a better performance in the future construction of ontologies.
D10.3.1 /Prototype

In the images below show some of the evolution of the QTO. The Figure 8.2 contains five topics detected from the questions. At that point, only a 46% of the questions were being used (the rest were still part of Guardia-root). The rest of the questions, 54%, had not been yet classified into any topic. It can also be seen that although topics such as *Defunciones* [deaths] or *Internamientos* [interments] had been located and named, other topics had not be named yet.

![Diagram](image)

**Figure 8.2**: Screenshot of the development of the QTO, only 46% of the questions had been used.

In the Figure 8.4, seven topics had been identified that covered the 74% of the total of questions. At that point, the team modelling the QTO checked the correctness of the specific questions being classified and moved them to their correct topic, if they had been incorrectly classified by the tool. Nearly 100 questions had been misplaced. This feature of OntoGen is very interesting as it allows the user to visualize the particular question and to deselect it from that topic and, thus, return it to the Guardia-root. The figure below, shows this functionality, on the bottom right of the picture you can see the question selected for analysis.
The team also analyzed the remaining questions contained within the Guardia-root, and generally, they considered that those questions had to be kept there because they were mainly referring to situations occurring during on-duty hours without any special relation to other topics already located.

Finally, when the main skeleton had been produced, the team developed the topics into sub-topics, taking into account the suggestions made by OntoGen based on keywords. In Figure 8.5 you can see that Oficina Judicial [court office] has two sub-topics (trial videotaping and prosecution), that Defunciones [deaths] has two sub-topics (corpse removing and autopsy), that Violencia Doméstica [gender violence] has also two topics (protection orders and restraining orders).
This is the current state of the QTO that still has to be evaluated and refined. The UAB team of legal experts is also producing a manual classification of the questions into several topics at the same time. This work, the classification, the evaluation and the refinement, is going to be performed over the following months.

8.9.3 Judgment Topic Ontology

A preliminary version of Judgment Topic Ontology (JTO) was constructed using OntoGen (see Figure 8.6) and we plan its refinements for the second part of the prototype Iuriservice (ongoing work for year 3). Some topics have been identified. However, the corpus used in JTO is significantly different (in size and preparation) and results regarding QTO evaluation can be very useful for JTO development and refinement. The current version of OntoGen does not support moving a sub-topic from a topic to another topic (with all the instances – judgments – with it).
Moreover, in collaboration with WP7 we have developed with the help of some documents regarding ontology learning preparation. These documents, the Ontology Requirements Specifications Document (ORSD), the Ontology Learning Requirements Specifications Document (OLRSD) and the Risk Analysis Document (RAD) contain the specifications on the goals of the ontology that wants to be built, tools and functional requirements, and critical points to take into account when learning the ontology using certain tools (within others). These documents can also contribute to building JTO and have to be refined with the experiences shown by the development of QTO. The DILIGENT methodology has to be applied as well.

8.10 Topic detection software module.

The practical purpose of topics, as described earlier, lies in the ability to automatically recognize them from the text. In practice this is achieved by several consecutive phases:

1. **Natural language processing on the text:**
   - *Input:* raw text
   - *Operations:* tokenization, lemmatization, stemming, morphological analysis, etc.
   - *Output:* lemmas with PoS tags

2. **Convert syntactically analysed text into features (relevance filtering):**
   - *Operations:* reducing the importance of semantically irrelevant words, identifying important sequences, increasing the importance of semantically relevant features.
   - *Output:* text represented as a bag-of-words

3. **Detect topics from text:**
   - *Operations:* reducing the importance of semantically irrelevant words, identifying important sequences, increasing the importance of semantically relevant features.
   - *Output:* semantic metadata in the form of probability assignments to different topics.
4. Perform semantic information retrieval:
   - **Operations:** semantic distance computation, etc.
   - **Output:** relevant answers, judgements, etc.

The current section of the document will concern itself with the operational aspects of the second and third steps – relevance filtering and topic detection. We will also describe the basic functionality of the present prototypic implementation of the technologies. For prototyping, we have applied Python, a powerful scripting language with powerful string-processing and associative array capabilities.

Both relevance filtering and text categorization are based on back-end databases. For this reason, it is highly inefficient to load the databases and initialize the system for each individual question: in the Q&A task, this would result in considerable latencies. Therefore, both methods are first initialized and then queried for a particular chunk of text.

The application program interface (API) is very simple, as shown in this example:

```java
// init the JEP interface
File pwd = new File(".");
jep = new Jep(false, pwd.getAbsolutePath());
// initialize the Concept - the interface between Java and Python
Concepts concepts = new Concepts("preguntas-model.pik");
// let Python know where to find the class
jep.set("input", concepts);
// start the Python script that will initialize everything
jep.runScript("prepare.py"); // (X)
//(X)
// classify a particular question
// it has been lemmatized, and each word is a feature
jep.eval("classify('acordar sobreseimiento unico acreedor')");
// print the concept membership probabilities
for(int i = 0; i < concepts.getConceptNumber(); i++) {
    System.out.println(concepts.names[i] + ':');
    System.out.println(concepts.probabilities[i]);
}
```

By the line marked with (X), the topic detection system has been initialized. With `jep.eval` we execute a classification query, and then read the detected concepts from the `concepts` object, which contains both the names and the probabilities. The relevance filtering can be executed before the question string has been generated.

9 Measurements.

Some measurements that apply to the Legal Case Study were defined in [25]. These measurements can be summarized at Table 9.1:

<table>
<thead>
<tr>
<th>Measure</th>
<th>Definition</th>
<th>Target</th>
<th>Reported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search-engine usage</td>
<td>Number of queries per day</td>
<td>Continuing growth</td>
<td>At the end of the project</td>
</tr>
<tr>
<td>Average Response time</td>
<td>Average time to respond to a user's</td>
<td>Not target at this stage</td>
<td>At the end of the project</td>
</tr>
</tbody>
</table>
The measures applied in the case studies will reflect the usage of the SEKT technology as implemented in the 3 case studies. These measures will be evaluated at the end of the project, i.e. during the last quarter of year 3.

In order to evaluate the current prototype, we plan to validate the improvement of introducing the concept of the semantic distance to find the question-answer that better match with the question-input. This evaluation will consist on the comparison of results obtained using a typical keyword based search engine and the results obtained on the application of keyword search in combination with ontology based search. To facilitate this task, we will use an audit system (integrated as part of the system) and also the multistage searching approach followed to design the case study architecture to switch on the respective stage.

At the moment, Expert System has been evaluated with two kinds of measurements: Efficiency measurement (average time to response to a user’s query) and Effectiveness measurement (precision of user’s query response).

9.1 Effectiveness measurements

This type of measurement compares precision of user’s query response between two types of search:

- Based on Keywords, Morphological Analysis and Synonyms.
- Based on Keywords, Morphological Analysis, Synonyms and Semantic Distance/Ontologies.

We use a corpus of 62 FAQs of Gender Violence domain. The ontology that models the domain is OPJK (82 classes, 484 instances, approx. 118 attributes).

There are to type of tests in this measurement:

- Questions posed by user with the same meaning as a target FAQ question.
- Questions posed by user with different meaning as a target FAQ question.

In both cases we define tests with 7 target FAQ and 5 user questions for each target FAQ (35 tests per case in total).

The next tables summarize the results of the tests, showing the success and failure percentage of the system response. We can see how the semantic distance ontology allows improving the search effectiveness.
9.2 Efficiency measurements

We developed a testing plan to evaluate the system efficiency when it uses (or not) caching. We execute 35 tests over a corpus of 164 FAQs, of Gender Violence and On Duty domains, with activated and not activated caching.

The testing machine features are:

- Development HP.
- Intel Pentium processor 1600 MHz.
- 1 GB RAM.

In the next tables we can see the results; those show how the caching reduces system response time in a percentage of 43.91 % (from 6.6 seconds to 2.9 seconds).

### Table 9.2: Same meaning effectiveness tests results

<table>
<thead>
<tr>
<th>Success</th>
<th>Keywords</th>
<th>28.57 %</th>
<th>Keywords &amp; Semantic Distance</th>
<th>45.71 %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failure</td>
<td>Keywords</td>
<td>71.43 %</td>
<td>Keywords &amp; Semantic Distance</td>
<td>54.29 %</td>
</tr>
</tbody>
</table>

### Table 9.3: Different meaning effectiveness tests results

<table>
<thead>
<tr>
<th>Success</th>
<th>Keywords</th>
<th>17.14 %</th>
<th>Keywords &amp; Semantic Distance</th>
<th>40 %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failure</td>
<td>Keywords</td>
<td>82.86 %</td>
<td>Keywords &amp; Semantic Distance</td>
<td>60 %</td>
</tr>
</tbody>
</table>

### Number Test | Total Response Time (ms) | Keyword Time (ms) | Semantic Distance Time (ms)
--- | --- | --- | ---
1 | 5608 | 1792 | 3816
2 | 9294 | 6179 | 3115
3 | 5117 | 5117 | 0
4 | 4326 | 3075 | 1251
5 | 11106 | 6329 | 4777
6 | 6620 | 2003 | 4617
7 | 9674 | 5298 | 4376
8 | 4596 | 1261 | 3335
9 | 5478 | 2524 | 2954
10 | 13560 | 4156 | 9404
11 | 2633 | 2363 | 270
12 | 3956 | 2514 | 1442
13 | 5508 | 4006 | 1502
14 | 6429 | 5728 | 701
15 | 6088 | 5437 | 651
16 | 8012 | 4877 | 3135
17 | 8762 | 6710 | 2052
18 | 4497 | 2033 | 2464
19 | 3815 | 3074 | 741
20 | 13029 | 3535 | 9494
### Table 9.4: Efficiency without caching

<table>
<thead>
<tr>
<th>Number</th>
<th>Test</th>
<th>Total Response Time (ms)</th>
<th>Keyword Time (ms)</th>
<th>Semantic Distance Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>1592</td>
<td>1502</td>
<td>90</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>4206</td>
<td>4156</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>4596</td>
<td>4596</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>2604</td>
<td>2594</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>2844</td>
<td>2794</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>1713</td>
<td>1683</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>4556</td>
<td>4536</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>1042</td>
<td>1022</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>2163</td>
<td>2143</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>3455</td>
<td>3415</td>
<td>40</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>1993</td>
<td>1993</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>2263</td>
<td>2253</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>3395</td>
<td>3385</td>
<td>10</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>2133</td>
<td>2123</td>
<td>10</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>4747</td>
<td>4747</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>3835</td>
<td>3805</td>
<td>30</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>3656</td>
<td>3656</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>1782</td>
<td>1772</td>
<td>10</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>2624</td>
<td>2614</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>2964</td>
<td>2934</td>
<td>30</td>
</tr>
<tr>
<td>21</td>
<td></td>
<td>2764</td>
<td>2764</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td></td>
<td>5498</td>
<td>3836</td>
<td>1662</td>
</tr>
<tr>
<td>23</td>
<td></td>
<td>4997</td>
<td>4697</td>
<td>300</td>
</tr>
<tr>
<td>24</td>
<td></td>
<td>1983</td>
<td>1983</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>3295</td>
<td>3285</td>
<td>10</td>
</tr>
<tr>
<td>26</td>
<td></td>
<td>1923</td>
<td>1863</td>
<td>60</td>
</tr>
<tr>
<td>27</td>
<td></td>
<td>2373</td>
<td>2373</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td></td>
<td>4276</td>
<td>4276</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td></td>
<td>2614</td>
<td>2594</td>
<td>20</td>
</tr>
</tbody>
</table>
## 10 Next Steps.

In this section, we describe the next steps in the development of the prototype:

- Evaluation of the results obtained from a comparison of the two kinds of search, keyword search and combined ontology and keyword search. We are developing new semantic heuristics for ontology search (heuristics that define semantic graph construction and distance). We are going to evaluate these heuristics and the improvements, which are expected be obtained.
- Integration of the current ontology in the last version of PROTON. The last changes to PROTON include some necessities related to the ontology of this case study.
- Integration with SIP. The first steps have been made in this direction. Some WP technologies will be used in this prototype, an example of this is the integration of the domain ontology detection using the technology of the WP1. Technologies related to the natural language processing (GATE) and ontology technology (KAON 2) will also be integrated as a pipelets of SIP. Therefore, the major effort in integration of technologies from the other workpackages will be completed for the development of the second part of the case study, the jurisprudence technology.
- In the final year of the project, we plan to integrate the Search & Browse technology (WP5) over jurisprudence data. Also we will use Alignment technology (WP4) to find which judgements associated with the FAQ are obtained as a result of the FAQ system. And we will use some results of WP1 technology, which is related to the crawling of judgments, and Judgement Topic Ontology (JTO) generation. We will use Visualization technology (WP5) to view Question Topics Ontology (QTO). We are integrating Legal Case Study Prototype (Expert and Jurisprudence Knowledge Systems) with SIP.
- We are working to improve the effectiveness and efficiency of Expert System. We are planning to run some user tests in 2006 (effectiveness improvement) with knowledge experts and final users. We will aim to reduce response time caching other subsystem parts (like ontology elements).
- We plan to design and develop the second system in Legal Case Study (Jurisprudence System).
- Finally, some improvements have been identified as a result of the computation of the paths between the concepts of the ontology, and the match with the question made by the user. Some of these improvements are aimed to improve the accuracy and performance of the system. Several algorithms can be used, currently we are studying an alternative based on A* algorithm [20].

### Table 9.5: Efficiency with caching

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>2534</td>
<td>2504</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>1642</td>
<td>1642</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>1502</td>
<td>1502</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>2995</td>
<td>2995</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>3404</td>
<td>3384</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>2424</td>
<td>2414</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>2925</strong></td>
<td><strong>2852</strong></td>
<td><strong>73</strong></td>
<td></td>
</tr>
</tbody>
</table>

---

80
that can be very appropriate for our purpose and very efficient computationally, but it implies transforming the ontology graph into a binary tree.

11 Conclusions.

In this deliverable, we have described the current status of the Legal Case Study prototype. The prototype has the following main features:

- It is designed to be accurate and technologically advanced, using NLP and Semantic Web techniques.
- It is designed to be efficient, extensible, customizable and scalable.
- It uses incremental searching as a process to narrow the desired FAQ set.
- It uses a variety of pluggable search algorithms.
- It integrates, or will integrate, some SEKT technology in its main subsystems (search, ontology and natural language processing).
- It defines a set of semantic heuristics that apply a semantic distance concept in text matching problems of specific domain.
12 Bibliography and references


