FAKULTÄT FÜR INFORMATIK

DER TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Applying lexical knowledge to improve search quality for a German legal information database

Laura Altamirano Sainz

FAKULTÄT FÜR INFORMATIK

DER TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Applying lexical knowledge to improve search quality for a German legal information database

Verbesserung der Suchqualität in deutschsprachigen Rechtsdatenbanken durch Anwendung von lexikalischem Wissen

Author:	Laura Altamirano Sainz
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Bernhard Waltl, M.Sc.
Submission Date:	April 15th, 2015

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, April 15th, 2015 Laura Altamirano Sainz

Acknowledgments

First of all, I would like to thank Prof. Dr. Florian Matthes for giving me the opportunity to write my Master's thesis within the Software Engineering for Business Information Systems chair.

A deep thank you to Bernhard Waltl for all the motivation and the extensive and great support received to carrying this research out.

Also, special thanks to each and every person who was somehow involved in the development of this work. To all my interviewees, to all the people I consulted: all your inputs and valuable feedback were of great help, I really appreciate the time you took to help me out.

Last but not least, thanks to my family: Mónica, Pepe, Max, Sofi and Christoph, for being always there for me and for backing me up in all my wishes.

Abstract

Search support mechanisms help to accomplish more effective searches in databases. Legal practitioners could truly benefit from these mechanisms since having good access to legal information is crucial for them to perform their job.

To conduct a search, one needs to formulate a query to introduce into a search system. This query can be composed of one or more terms but figuring out these terms is not always a straightforward task. Lexical knowledge refers to the known information about a word represented through its related terms. If we take a search term and retrieve its lexical information, we might have an opportunity to enhance already known search support mechanisms with it. This information can offer new options to the users to reformulate their queries.

This thesis focuses on integrating the lexical knowledge extracted from a German lexical database into a search support mechanism for a German-legal-information database.

The scope of this work is narrowed to legal documents due to the advantages acquired from them since these documents tend to be uniform and to lack of distracting colloquialisms. Nonetheless, this does not shadow their relevance or vastness.

A needs assessment is performed to figure out the real user requirements as part of the user-centered design approach. Also, an investigation about currently used search support mechanisms is carried out to identify which ones are common in the legal domain and which ones can be improved by the integration of the lexical knowledge.

Based on the gathered information, a search system is built to express the lexical information as a query reformulation tool (i.e. a search support mechanism). For this, three lexical relations are selected to be integrated in the system: hypernymy, hyponymy and sibling terms. Users can interact with this information and expand or narrow their queries in the system. Moreover, GermaNet database is chosen to serve as the lexical source for German vocabulary and searches are performed on a set of German law documents.

Finally, the system implementation and favorable empirical results are reported in this thesis.

Contents

Part I.

Introduction

1. Introduction

1.1. Motivation

With increasing stored datasets in the past years, there is clearly a growing need to search properly within databases and in the most convenient way possible for uncountable reasons. An opportunity can be detected here to assist the users of such systems since naturally, manual search becomes unfeasible.

Searching is a very commonly performed task, even in everyday lives; therefore, the importance of improving the experience for the user as well as the search process itself to retrieve the correct information becomes really high, as well as, a great challenge.

Nowadays, many search engines bridge the gap between a user feeling lost while searching in a database and a user finding what he is looking for. However, there are still chances of improvement and areas that have not been fully explored which might offer a great help for the search process and perhaps provide more effective tools to search for the users.

Within those areas, we can spot the usage of lexical knowledge. Information about a word and its relations to others is comprehended within this knowledge. If we think that this information can be integrated in a search engine, then the hypothesis of supporting users throughout their searches by showing them relations of words gathers strength. These relations might be exactly what they are looking for and the search would be done more efficiently.

This thesis focuses on discovering how the lexical information can be integrated in search systems to provide intuitive interfaces, as well as whether this would be actually useful, specifically, for a legal-database user. A system is built upon the results of an empirical study and evaluated accordingly.

1.2. Research scope

The scope of the thesis has been reduced to German legal databases to achieve a more accurate searching space. This in the sense that all legal texts own a specific style of writing which results in uniform documents in most of the cases.

Moreover, the number of laws and regulations in Germany is constantly increasing, as well as the legal cases, which are often the base for legal resolutions. Thus, we obtain a rather sufficient search space, plus we can highlight the relevance of performing a proper search within this kind of documents. This will be looked at closely in the following chapter.

1.3. Research questions

The following are the challenges that will be tackled during this research:

1. ***How can search quality be improved by lexical information for a legal database?***
   This question aims to answer how and if legal-database users can actually be supported by lexical information while performing a search to fulfill their individual tasks. Also, it targets to find out how their searches can be done more efficiently by providing them with a support tool to help them find the information they need in a faster way.

2. ***What mechanisms and methods are common in legal databases?***
   It is planned to investigate mechanisms and methods in databases, particularly which ones are often used in the legal domain. The purpose of doing this is to have a broad selection, to notice why they are used for legal databases and to evaluate which of them can be improved by the usage of lexical information.

3. ***Which search mechanisms and methods can be enhanced by lexical information and how?***
   This question concerns which tools of the selection can be integrated with lexical information and, more importantly, how this information should be presented to the users to genuinely support them.

4. ***How can a implementation for a support search mechanism integrated with lexical knowledge look like?***
   This last research question targets the implementation of a search support mechanism and the evaluation of the research, to figure out whether legal-database users find the lexical information, suggested as a support tool to perform their searches, useful or not.
   Also, the information gathered will help to decide if the lexical information is really able to support query reformulation and enrich the searchers' experience.

1.4. Initial hypotheses

Retrieving incorrect results during a search might confuse the users and put them in an uncomfortable situation towards thinking of another way of searching for what they need; query reformulation tools can be of great help in these cases (Hearst, 2009). Based on this, the main hypothesis for this work is that the lexical information, presented as a search tool, helps to organize the users' ideas when they need to perform a search.

Another hypothesis is that integrating lexical information into a search system will enhance search quality since the whole search will be done more effectively.

To exemplify the first two hypotheses, let us imagine the following situation: Somebody needing information types a word to search for in a search system. The results given do not satisfy the user because actually he meant another word. If we show that user the lexical information of the word he typed, there are high chances that he will easily spot what he meant, as these information will be related words which will serve as a guide to reformulate the initial query.

Finally, the last hypothesis is that some search mechanisms exist that can be enhanced with lexical information. For instance, adding lexical relations of the word, such as synonyms, to reformulate the query and expand it. Nonetheless, it is evident that not all mechanisms will be able to be incorporated with such knowledge; however, a way to show this information to the users must be deduced.

1.5. Reader's guide

The following chapter will present, first of all, a theoretical analysis giving a synopsis of the most relevant related work and the most important concepts for this research (chapter 2).

Then, chapter 3 will show the approach followed to find a proper solution, plus the needs assessment performed and a comparison between existent search support mechanisms in a selection of five legal databases. The system design is also comprehended within this chapter.

The consecutive chapter will present a detailed description of the system implementation, defining all of its features (chapter 4).

With regards to chapter 5, the empirical evaluation for the system will be stated, as well as the faced limitations.

Lastly, chapter 6 will conclude with the outlook of this work plus the answers to the research questions after developing this investigation.

2. Theoretical analysis

2.1. Lexical knowledge

Lexical knowledge represents all the known information about words and the relations between them (O'Hara, 2005). In the artificial intelligence approach, this is often called "ontology" and pretends to construct a model that explains the relations of the entities (Pustejovsky & Bergler, 1992).

2.1.1. Ontologies

According to Onyshkevich and Nirenburg (1995), an ontology is a model of the world, a body of knowledge of the world organised as a taxonomy. Please refer to Figure 2.1 for a simplified example.

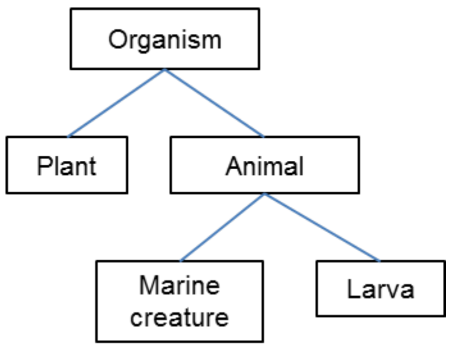


Figure 2.1.: Ontology fragment extracted from WordNet.

In Figure 2.1, if we focus in the word "animal", we can find out that an "animal" is an "organism", or that a "marine creature" and a "larva" are animals. This is acquired information that in this example is rather simple; nevertheless, it can be greatly enlarged by adding more related words to the taxonomy.

What is important to notice, is that we just need to know one word to extract more information about it from the ontology, precisely the idea of integrating this kind of information to a search engine: Getting more information from what we already know.

In other words, all this ontological knowledge might be useful for the searchers but they might not think about these relationships in the first instance when they are looking for certain information. Therefore, if a system can suggest it, their search time can be reduced and their progress in their tasks increased.

Furthermore, ontologies are domain specific. For example, if we take the word *System* and we look for its lexical relations in a biology-domain ontology, we will get relations to the digestive system or to the nervous system, for instance. Whereas, in a computational-domain ontology, the relations will be for example, to the operating system.

2.1.2. Lexical databases

There exist many lexical relations between words; however, this thesis focuses on a set of them due to the fact that they are the most relevant, and the technologies used for the development of the system work with them. *(See Table 2.1 for a list of the lexical relations involved).*

All this lexical information must be stored in a database that we can query with a certain word so that we can obtain the related words. For this work, WordNet and GermaNet technologies are used as the databases:

WordNet
WordNet is an online lexical reference system from Princeton University for English, where nouns, verbs and adjectives are organized into synonym sets (synsets) (Fellbaum, 1998. Miller, 1995; Miller, Beckwith, Fellbaum, Gross, & Miller, 1990). The relations connecting the 117,000 synsets that are contained in this database are conceptual-semantic and lexical (Princeton University, 2010a).

This thesis is focused on the German language; however, WordNet was used as a base when the work started since an academic research license is required to use the German lexical database (i.e. GermaNet) and time was needed to obtain it.

GermaNet
GermaNet, developed at the University of Tübingen, is a lexical semantic net for German. This net aims to model the German's base vocabulary. Its framework is compatible with the Princeton WordNet since the same technology for the database format and the database compilation is used, as well as the WordNet interface with extensions where necessary (Hamp & Feldweg, 1997).

GermaNet version 9.0 release 2014 contains 93 246 synsets (University of Tübingen, 2014). A synset is a set of lexical units (i.e. words) and represents the relation of synonymy (Henrich & Hinrichs, 2010). *(See Table 2.2 for a summary of GermaNet contents).*

Similar to WordNet, GermaNet deals with two types of semantic relations: conceptual and lexical. The former one refers to relations between two semantic concepts (synsets) (e.g. hyperonymy) and the latter one refers to relations between two individual lexical units (e.g. antonymy). To exemplify the lexical relations that GermaNet provides, let us look at the word *richtig* (right) extracted from the database in Figure 2.2.

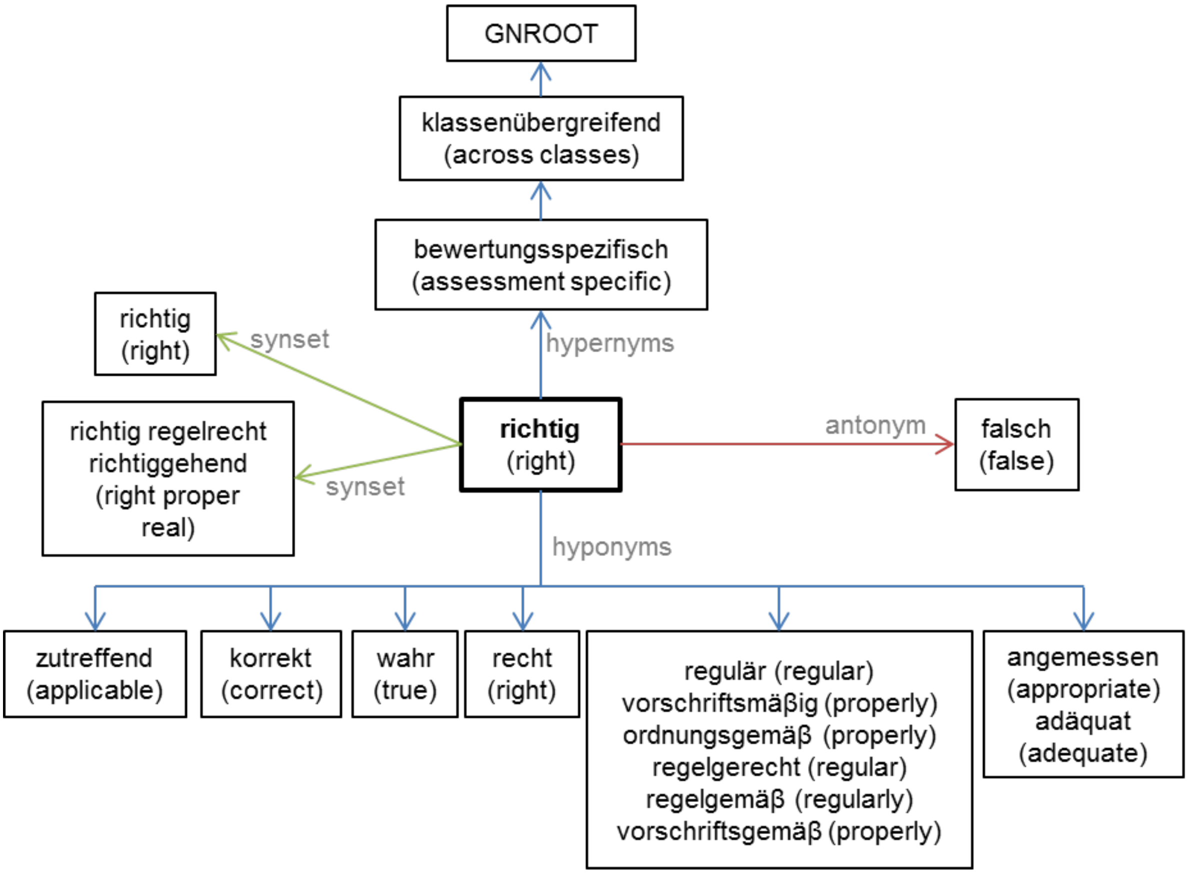


Figure 2.2.: Lexical relations example extracted from GermaNet. Note that words grouped in the same box are words contained in the same synset and that GNROOT is the top node of GermaNet, meaning that there is no hypernym above this element.

Table 2.1.: Lexical relations (Princeton University, 2010b).

Relation	Description
Synonym	X and Y are interchangeable in some context without changing the truth value of the preposition in which they are embedded. e.g. a *car* is synonym of an *automobile*.
Hyponym	X is a hyponym of Y if X is a (kind of) Y, e.g. a *convertible* is a hyponym of *car*.
Hypernym	Y is a hypernym of X if X is a (kind of) Y, e.g. a *vehicle* is a hypernym of *car*.
Meronym	A member or a constituent part of something. X is a meronym of Y if X is a part of Y, e.g. a *rim* is a meronym of *wheel*.
Holonym	The name of the whole of which the meronym names are part. Y is a holonym of X if X is a part of Y. e.g. a *wheeled vehicle* is a holonym of *wheel*.
Troponym	X is a troponym of Y if X is to Y in some manner, i.e. X is a particular way of Y. e.g. *test driving* is a troponym of *driving*.
Coordinate terms (siblings)	are nouns or verbs that have the same hypernym, e.g. a *motorcycle* is a coordinate term of *car* (both share the hypernym *vehicle*).
Derivation-ally related forms	Terms in different syntactic categories that have the same root form and are semantically related, e.g. a *driveway* is a derivationally related form of *drive*.

Table 2.2.: Statistics of GermaNet Version 9 contents (Waltl & Matthes, 2015).

Content	Count	Percentage
Synsets	93 246	-
Lexical units	121 810	100 %
Conceptual relations	105 912	-
Nouns	93 631	76.87 %
Verbs	14 331	11.76 %
Adjectives	13 848	11.37 %

2.2. German legal information

As mentioned before, the research scope of this thesis is set on German legal databases.

"The number of laws in the Federal Republic of Germany is overwhelming and growing steadily" (Deutscher Bundestag, February 2013). This highlights the importance of conducting a proper search within these documents, since only the laws themselves (without taking into account cases or legal journals, for instance) are already an extremely wide search space.

Moreover, this information is frequently revised for innumerable reasons. For instance, revising it to build up a case or perhaps to motivate a case resolution. Thus, it is truly relevant information and in many cases, being able to perform an efficient search would be very satisfying for the searchers.

What is more, legal information offers a standard format, meaning that the organization of the document contents is done in a consistent way. This allows one to work more precisely with the data and, particularly, to focus on solving this thesis' challenges without data-presentation distractions which are out of the scope.

2.3. Search quality

Crosby defines quality as conformance to requirements (1979). On the other hand, Hearst (2009) points out that aiding users in the expression of their information needs and in the formulation of their queries are jobs of the search user interfaces. Therefore, to meet the requirements, the produced interface must help the users in a way to orientate them so that they can retrieve what they need from the database.

Moreover, "user studies strongly suggest that standard search techniques have to be improved in order to meet legal particularities" (Schweighofer & Geist, 2007). In Table 2.3 a compilation of the search tools or mechanisms that are going to be studied is presented. These are based on the *Search user interfaces* book (Hearst, 2009), plus mechanisms observed in search systems. They are divided into three main categories that cover the basic components of the information retrieval process (as marked in Figure 2.3):

1. **Query formulation or query specification** – referring to mechanisms that support the creation of the query to search for within the database (before the search is performed).

2. **Query reformulation** – after the query has been submitted and the results have been produced, query reformulation tools help the users to find a way to expand or narrow their search to produce better results according to their needs (after the search is performed).

3. **Integration of navigation of the results with search** – these refer to mechanisms that provide filtering of the results or categories to present the results in a more organized way for the convenience of the users (after the search is performed).

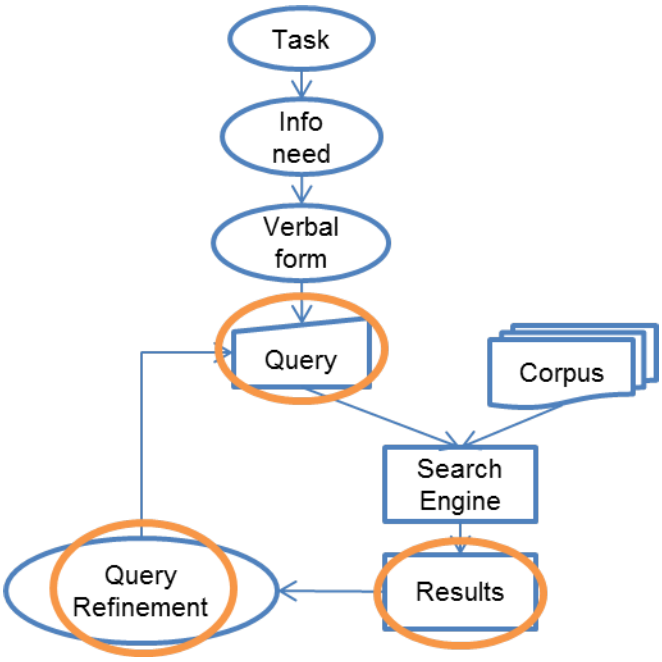


Figure 2.3.: The classic information retrieval model, augmented for the web. Adapted from (Broder, 2002)

Table 2.3.: Search support mechanisms (Based on the *Search user interfaces* book (Hearst, 2009) and mechanisms observed in search systems).

Search support mechanism	**Description**
Query formulation/specification	
Query specification using Boolean or other operators	These operators allow a more careful formulation of queries. For instance, the use of Boolean operator "AND" in a query to specify two (or more) required terms that should be present in the results.
Continued	

Search support mechanism	Description
Wildcard characters	These characters allow expanding the search query with any word(s) or character(s) that we might not know or simply want to create a expression that can match all different possibilities. For example, the character "*" in the query "president * Lincoln" will be replaced by the president's name because that is the possible word that will be found in documents.
Segment search/ query specification using command languages	Some systems allow the users to include specific commands in their queries. For example, the command "DATE IS" can be used to specify in the query that we only require matches with certain date.
Autocomplete	The autocomplete mechanism presents the users with drop-down menus that show queries that were previously issued whose prefixes match what the user is typing in the input field. This helps to specify the query faster or to show possible words that complete the input the user is introducing.
Parametric search /scoped search	Here, users must specify a certain number of attributes that they require their results to match. For example, when looking for a book in a search system, one can specify the author or the year before issuing the query. However, queries can become so specific that they might often lead to empty result sets.
Query reformulation	
Relevance feedback	This is an interactive mechanism that allows users to skim through the results and mark those ones they find relevant. The system takes the feedback into account to update the query and therefore, update the results.

Continued

Search support mechanism	**Description**
Spelling suggestions and corrections	This mechanism suggests spelling corrections for the query or directly corrects it.
Automated term suggestions / Query expansion	Tool that suggests alternative words and phrases, commonly as a hyperlink, to reformulate the query (after it has been issued). These words or phrases might expand or replace the query. *Query refinement suggestions* – This tool is a type of automated term suggestions. However, suggestions here refine the query, for instance, to a certain context. For example, if a user types the ambiguous word *"Apple"*, then the system may suggest *Apple the Fruit* or *Apple Computers* and the context would be specified. Some engines offer these suggestions with the *"Are you looking for?"* question.
Suggesting popular destinations	Offering popular destination websites for a given query.
Showing related articles	Offering *More like this* article links.
Integration of navigation of the results with search	
Categories for navigating and narrowing	This mechanism offers help for navigation by narrowing down the content by adding a specific category to the query. It is similar to term suggestions tool but here the query is not replaced rather, a category is selected and results in this category are the only ones presented. In other words, the query is not reissued. Categories can be specified by studies in the users behavior and only one category can be seen at a time. For example, a website that sells books might organize their products according to the type of book (i.e. fiction, romance, kids, etc.). And when one of these categories is selected, only items belonging to it will be displayed.

Continued

Search support mechanism	Description
Categories for grouping search results	Here, the results are grouped, assigned and presented in categories. This might make easier skimming through them to find what the user is looking for. Results are labeled with a category and many categories can be seen in the same interface.
Categories for sorting and filtering search results	Users are presented with categories that help to reorder the results according to certain attributes' values. For example, sorting the results by alphabetical order. Also, filters are shown to be able to narrow down the hits.
Organizing search results via table-of-content views	This mechanisms organizes the results in a hierarchy or a tree structure. For example, showing a table-of-contents for a book, where the user can navigate through chapters and sections. This tool is still not commonly used.
Faceted navigation	With faceted navigation, results are not assigned to single categories because this produces a conflict: Documents not only discuss one topic but, more commonly, many. Therefore, this tool offers a set of categories as well as attributes. The categories and attributes will be grouped by feature types, i.e. facets. Users can select which attribute(s) shall delimit the results. This mechanism is also known as guided navigation.
Navigating via social tagging and social bookmarking	This tool allows users to assign one or more category labels (i.e. tags) to web pages.
Other search support tools	
Highlight query terms	This mechanism marks the query terms in the result list to let the user know which concept was found in each listed document/result.

Continued

Search support mechanism	Description
Number of matches	The number of results is visible to the user.
Font size change	Here we can augment or reduce the font size of the website.

2.4. Related work

Divoli, Hearst, and Wooldridge (2008) made a study involving biologists, where they inquired them whether they would find useful ontological relations for gene/protein names incorporated into bioscience literature search user interfaces or not. Their results show that the majority of the biologists involved would like to see this type of information. They followed the user-centered design approach and performed two questionnaires. One to find out which kind of information participants would like to see, and the second one, after designing 4 scenarios and adding them to the group's search engine, to find out with which one the participants felt more comfortable with.

This work is absolutely relevant for this thesis, since also here the user-centered design will be followed. Plus, knowing that ontological knowledge is helpful for other disciplines, such as biology, serves as a base to use it in the legal domain.

Another project pertinent to mention is the CIRI (Concept-based Information Retrieval Interface) system (Airio, Järvelin, Saatsi, Kekäläinen, & Suomela, 2004), which is based on a three level model that allows one to express ontological relations. In this system, users are able to select concepts from available ontologies and browse them. Additionally, they can choose search engines and databases to perform their searches. The query is built directly from the ontologies, which is an advantage as the system will search directly from the lexical relations to retrieve the results.

Bast, Chitea, Suchanek, and Weber (2007) also presented an ontological-related efficient system: ESTER (Efficient Search on Text, Entities and Relations). This proposal was built for combined full-text and ontology search. In their paper, the authors highlight the importance of performing semantic searches and provide the system with an entity recognizer, which assigns words or phrases in the corpus to the ontology entities they refer to. Moreover, their system is able to perform prefix search and suggest possible semantic interpretations while executing the one which is more likely to be searched.

We should note that CIRI allows the user to browse the ontologies, which lets the users know the lexical relations of the word(s) they are searching for, whereas, ESTER integrates the ontologies in a very different way, which makes them invisible for the users. This latter approach is observed to be more commonly implemented than the first one. The following proposals are also focused on this direction.

A model for extracting information in the legal domain was developed in the University of Vale do Rio dos Sinos - UNISINOS (Araujo, Denis Andrei de, Rigo, Müller, & Chishman, 2013). This model makes use of semantic information and integrates linguistic information obtained from studying legal documents. They created a Brazilian-legal-domain ontology and a set of knowledge acquisition rules; results indicate good prospects when applying them to search within a set of documents. It is relevant to remark that they believe that it is through ontologies that computational tools for legal information retrieval can be improved.

Smith (1997) described "how a lexically designed search engine in a domain specific database, such as a large database of legal cases, can provide better relevance ranking than best match search engines." The system built demonstrates that lexical structures can considerably improve information retrieval in legal databases and Smith states the high potential of computational lexicology for artificial intelligence as well as for the law domain.

Saravanan, Ravindran, and Raman (2009) also mention the use of ontologies in their paper. They created a framework that sets the ontological information between the user interface and the database (built upon legal information) and deduced that ontologies enable inferences that can ensure effective information retrieval. According to their results, ontology-based searches generate much better results than traditional methods.

Another important project is the one developed by Dini et al. (2005). The LOIS (Lexical Ontologies for legal Information Sharing) project aims to build a semantically enriched and multilingual terminological database that follows the WordNet framework (Fellbaum, 1998. Miller, 1995). These authors stated that if knowledge is modeled by using ontologies or enhanced thesauri, then the ability to extract and exploit information from documents is enhanced by the explicit links that can be established among related items. LOIS is composed of two modules: a lexical database and a legislative one. The project's goal is to improve legal information retrieval and connect a WordNet with six European languages, which allows users to investigate different issues such as multiple senses of the words (due to different legislative sources) or compare between national legal systems.

On the search mechanisms domain, Jothilakshmi, Shanthi, and Babisaraswathi (2013) performed a survey on semantic query expansion. They define query expansion as "the process of reformulating a seed query to improve retrieval performance in information retrieval"; to put it differently, to add words or phrases to the query. Their research lets us know that this mechanism allows overcoming the inaccuracy of a query formed by a few words. Therefore, this mechanism is rather promising for search systems.

Schweighofer and Geist (2007) work also with query expansion. Their project aims to develop a methodology to improve Boolean search by using lexical ontologies and user relevance feedback. The former refers to query expansion with a lexical ontology; and the latter refers to collect relevance information from documents from an issued query to issue a second one. Results when the system expands the query with synonyms (which are established by the system) were quite good. It is noteworthy that their results also "revealed that users may not be able to find a proper search term for the knowledge base as common language may use a different term," which is exactly what this thesis aims to solve.

In summary, the ontology domain has been explored in many ways to be combined with databases. Most of prior research pursue to integrate the lexical knowledge in the background of the search systems, making it invisible to the user. On the other hand, there is also some work that requires the interaction of the user with the ontologies. Both approaches have produced promising results and consider that ontological information can improve the search process. However, there are still questions to be answered in these fields, especially for legal texts. Moreover, query expansion and other search support mechanisms, can be explored in different ways and other means to enhance them with lexical knowledge may be found.

Part II.

Thesis contribution

3. Approach

3.1. Main objectives

To begin with, let us summarize the main objectives of this thesis:

1. **Find out existing search support mechanisms in legal databases and investigate which mechanisms can be integrated with lexical information.**
   A comparison between search support mechanisms in existent legal databases must be done to be able to analyze which ones benefit from the integration of lexical knowledge.

2. **Assess the needs of the user.**
   This is a very important step towards the discovery of what the targeted users really need to have and to know what are their thoughts about this kind of support.

3. **Build a system that supports legal database users.**
   All knowledge acquired must be expressed in a search system that integrates a legal database and lexical knowledge.

4. **Present the lexical information as a search support mechanism in the system.**
   This task aims to integrate the lexical information in the front-end of the search engine, where users can interact with it. Therefore, a way to implement it as a search support mechanism must be found.

5. **The support tool must be intuitive for the user.**
   For legal practitioners, good access to information sources is crucial (Matthijssen, 1999). Therefore, search user interfaces should orientate the users to help them get the information they need and not confuse them. Thus, the lexical information, which can become complex or abundant to a certain point, should be presented as simple as possible in the system.

6. **Evaluate the system.**
   As in any other research, the resulting system must be evaluated accordingly.

3.2. Research method

This thesis follows Hevner, March, Park, and Ram (2004) guidelines for design-science in information systems research. Figure 3.1 lists the seven guidelines.

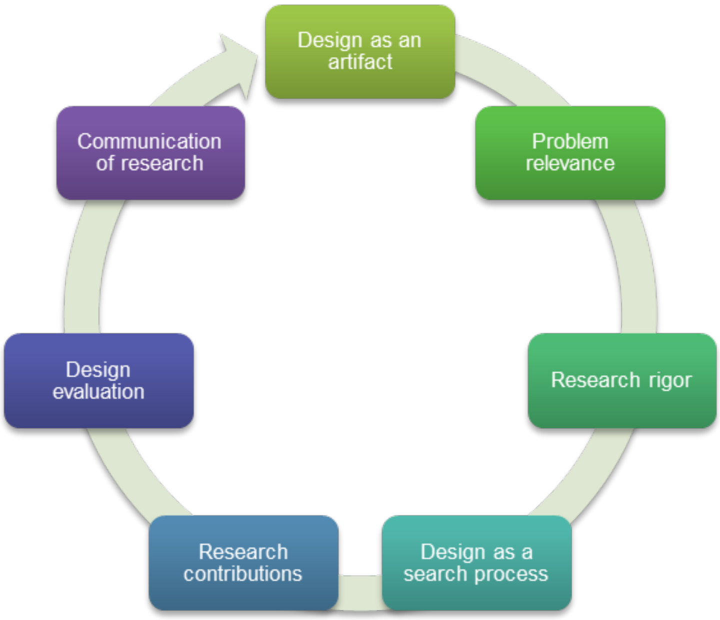


Figure 3.1.: Guidelines for design science in information systems research. (Hevner, March, Park, & Ram, 2004)

This research approach tackles the problem by designing the solution as an artifact, which in this work is a search system. Moreover, the problem relevance is defined (section 1.1), which describes the importance of properly searching in legal databases.

The research rigor refers to all the related work in the field which can be found in section 2.4. Regarding the design as a search process guideline, this thesis is based on the related work encountered to build up a research contribution.

Then, the design evaluation will be performed and the communication of the research alludes to the finalization of the written report, i.e. this thesis.

The purpose of following this methodology to approach this investigation is to make sure that all aspects are covered to derive in a complete research.

3.3. Search support mechanisms in legal databases

Five legal databases were selected to analyze which search support mechanisms are common in this field and which ones can be improved by using lexical information:

JURION
JURION is a customizable environment to work with a wide variety of German legal material (Wolters Kluwer, 2014a). *JURION Suche* module is a search system that provides fast and accurate access to the legal information (this module was the one analyzed and compared to the other legal databases).

beck-online
This database is also for German content (Verlag C.H.BECK oHG, 2014a) and one can search through legal documents such as books, magazines, laws, among others.

LexisNexis
LexisNexis provides access to legal documents worldwide. It is specifically designed for professionals in the legal, risk management, corporate, government, law enforcement, accounting and academic markets. Along with their free version, they offer an advanced and an academic version (RELXGroup, 2015a).

FindLaw
FindLaw is a portal that provides legal information and marketing solutions for law firms (Thomson Reuters, 2015a). As well, it makes it possible to look for legal professionals within their database.

LEXinform
DATEV's LEXinform is a database for national (Germany) and international tax, business and civil law, as well as business administration material (DATEV, 2015a). Analysis of this database was conducted without a user account.

Table 3.1 presents a summary of the features observed in each of the databases. Please note that the search mechanisms that are not ticked in the table were either not found as a feature of the database or not encountered in the versions analyzed nor within the available information in their websites. For an explanation of each of the search mechanisms please refer to section 2.3.

Table 3.1.: Search support mechanisms in legal databases

Search support mechanism	**JURION**	**beck-online**	**LexisNexis**	**FindLaw**	**LEXinform**
Query formulation/specification					
Query specification using Boolean or other operators		✓	✓(*Ac*)		
Wildcard characters		✓			
Segment search/query specification using command languages		✓	✓(*Ac*)		
Autocomplete	✓	✓	✓		
Parametric search /scoped search	✓	✓	✓		✓
Query reformulation					
Relevance feedback					
Spelling suggestions and corrections		✓	✓		
Automated term suggestions	✓ (Related concepts)				
Query refinement suggestions	✓				
Suggesting popular destinations		✓ (Top-paragraphs)	✓(*Ac*)		
Continued					

Search support mechanism	JURION	beck-online	LexisNexis	FindLaw	LEXinform
Showing related articles					
Integration of navigation with search					
Categories for navigating and narrowing					
Categories for grouping search results					
• Origin/type of publication (Forms, magazines, books, comments, blogs)	Presented as facet	Presented as facet	✓	✓	
• Field	Presented as facet	Presented as facet	Presented as facet(*Ad*)		
Categories for sorting and filtering search results					
• Relevance criteria	✓	✓	✓(*Ad*)		
• Date		✓			
Organizing search results via table-of-content views					
Faceted navigation					
• Origin/type of publication (Forms, magazines, books, comments, blogs)	✓	✓	✓(*Ad*)		
• Field	✓	✓	✓(*Ad*)		
Continued					

Search support mechanism	**JURION**	**beck-online**	**LexisNexis**	**FindLaw**	**LEXinform**
• Timeline	✓		✓(*Ad*)		
• Court			✓(*Ad*)		
• Jurisdiction (i.e. Federal or State)			✓(*Ad*)		
• Subscription			✓(*Ad*)		
• Publication status			✓(*Ad*)		
• Citations information			✓(*Ad*)		
Navigating via social tagging and social bookmarking		✓			
Others					
Highlight query terms	✓	✓	✓(*Ad*)		
Number of matches	✓		✓	✓	✓
Letter size change	✓	✓			

Ac. LexisNexis Academic.
Ad. LexisNexis Advanced.
Please refer to appendix A for graphical information about where these elements were found within their respective websites.

In general, many of these search support tools are common in legal databases, such as the mechanism that allows the user to sort out the results by specifying the origin or type of publication. Also, tools are sometimes presented in different ways such as the *Showing related articles* mechanism is presented in JURION as *Related concepts* but the idea remains the same. However, not all of these features can be enhanced by lexical knowledge.

As a preliminary step, the following mechanisms were selected as possible to be enhanced:

1. **Autocomplete** – By looking at the piece of the word that is being typed in the input field of a search engine, the system can not only suggest completed words that start with these stems/prefixes, but also words that are derivations of the possible completed words. This way, users can have suggestions to specify better their queries since the beginning of the search process.
   Lexical relations like hyponyms or derivationally related terms can be part of the suggestions. Hyponyms can result very beneficial and specially for German language, since many compound words share one of the component words with their parent term. E.g. *Fach* (Subject) – *Fachbereich* (faculty).

2. **Automated term suggestions/Query expansion** – Lexical relations seem promising in this mechanism since the suggestions for the users can be lexically related words to the queried one and not only based on query logs or top-ranked results, for instance. The hypernymy relation offers a term broadening; therefore, the query could be replaced by the hypernym and the search would be widened.

3. **Query refinement suggestions** – Currently, this mechanism suggests terms that add context to the query. These terms are not necessarily lexically computed rather retrieved from a certain domain where the term might fall into.
   The hyponymy relation goes one step lower in the ontology tree of a term, i.e. a refinement of the original term. Consequently, this lexical relation could be very beneficial for this mechanism as the context can be derived from the hyponym.

4. **Categories for navigating and narrowing** – Hyponyms can serve as categories of the term the user is looking for. Then, results can be divided by each category. However, since this is not a query reformulation tool, the search should be performed then by matching also the term's hyponyms, so that the results can be categorized.

5. **Categories for grouping search results** – Similar to categories for navigating and narrowing, results can be categorized by hyponym terms. Likewise, matching should be also performed by searching for the hyponyms.

On the other hand, results could be also labeled/categorized with the hypernym terms, e.g. labeling a document about *dogs* as part of *canines*.

6. **Organizing search results via table-of-content views** – The ontological relations build an ontological tree by themselves, presenting this non-linear tree to the users might help them to organize the idea of where the word they are looking for belongs to.
For example, if the hypernyms and hyponyms are presented, users could navigate the ontology tree upwards or downwards to expand or narrow the search. However, similar to the two categories before, results should also match the hypernyms and hyponyms so that they can be categorized.

7. **Faceted navigation** – Hyponyms of the input word can be another way to organize or filter the categories that the results might fall into. This way, users could filter the results by selecting the hyponym.
Since this category divides the results, search for hyponym terms should also be performed to be able to classify the matches/result list.

3.4. Needs assessment

"Systems are built to help people work better. They cannot be built well without understanding how people work" (Holtzblatt & Beyer, 1997). Therefore, the goal of performing a needs assessment was to better understand how the target users of the system think and whether they would find the proposal useful. Also, performing this study allowed one to find crucial points that should be taken into account for the whole thesis. Please find the interview script in appendix B.

This assessment involved six people, all experienced in the law domain and all use a computer to perform their job more than 50% of their time. During the interview, they also mentioned that they review legal literature very often.

All participants agreed that legal information databases will become more important in the future and that formulating the query in the search system is one of the most important procedures while searching. Answers were not so uniform when participants were asked whether they think that recommendations (offered by a system) support the efficiency and effectiveness of their search. While three agreed completely, the other three were doubtful about the type of recommendations.

Interviewees were also presented with four types of recommendations or tools to perform searches. 100% of the participants answered that the selection of information source is completely relevant for the process (which was expected as the results of the database comparison showed that this is a very common search support mechanism in legal databases). Whereas, the autocomplete tool was not as favored.

Three imaginary situations were selected to extract the ontological information of the main word involved from WordNet database and to request the target users to rate whether they would find each relation to the word useful to give motivation or foundation to each of the cases.

In regard to the first situation, participants were asked the following question: *Imagine you are searching for the term "Abortion", what kind of information would you like the system to suggest? (Scale from 1 to 5, 5 being the highest)*. Figure 3.2 shows the interviewee's ratings to each of the lexical relations presented.

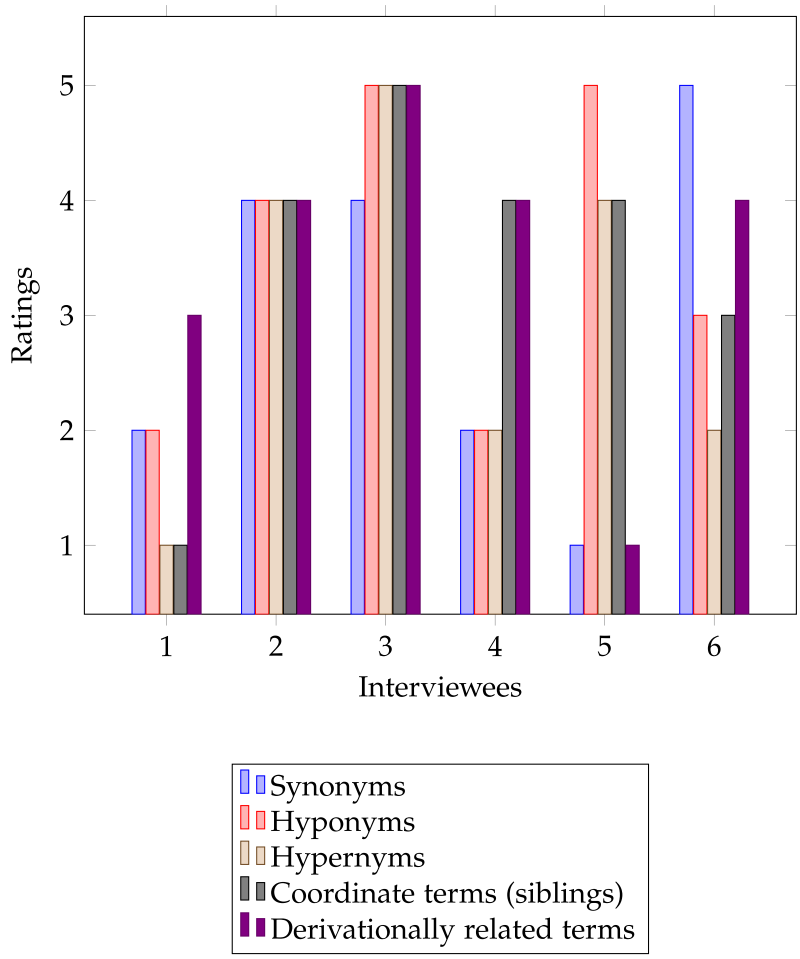


Figure 3.2.: Ratings to the question: Imagine you are searching for the term "Abortion", what kind of information would you like the system to suggest? (Scale from 1 to 5, 5 being the highest).

Table 3.2 summarizes the average ratings. Here we can see that participants are more interested in seeing hyponyms of the word or its siblings and derived terms rather than synonyms or hypernyms. However, it is noticeable that there is not a big numerical difference between the ratings.

Table 3.2.: Summary of average ratings for the first situation.

	Average rating	**%**
Synonyms	3	50%
Hyponyms	3.5	58.3%
Hypernyms	3	50%
Coordinate terms (Siblings or related terms)	3.5	58.3%
Derivationally related terms	3.5	58.3%

With respect to the second situation *(See ratings in Figure 3.3)*, we can identify, in the average ratings (Table 3.3), that the target users would like to see troponyms as part of the suggestions. Although, they found coordinate terms (related terms) also relevant.

Table 3.3.: Summary of average ratings for the second situation.

	Average rating	**%**
Troponyms	4.5	75%
Coordinate terms (Siblings or related terms)	3.5	58.3%

The third case was a bit controversial between the participants. Answers (See Figure 3.4) were considerably polarized. Table 3.4 shows the arithmetic mean of the ratings for the lexical relations involved in this question. These results were really interesting since, according to the answers, the presented relations appear to be useful only in certain law domains. While in others, they can be useful only in giving a broader overview to the lawyers and making them better understand the case but not to add facts to it.

Holonyms' average rating in the third situation was only of 44.4%. One of the participants, who gave this relation a rating of only one, mentioned that in this case, law is only interested if the arm can be healed, how long it would take or if it is a

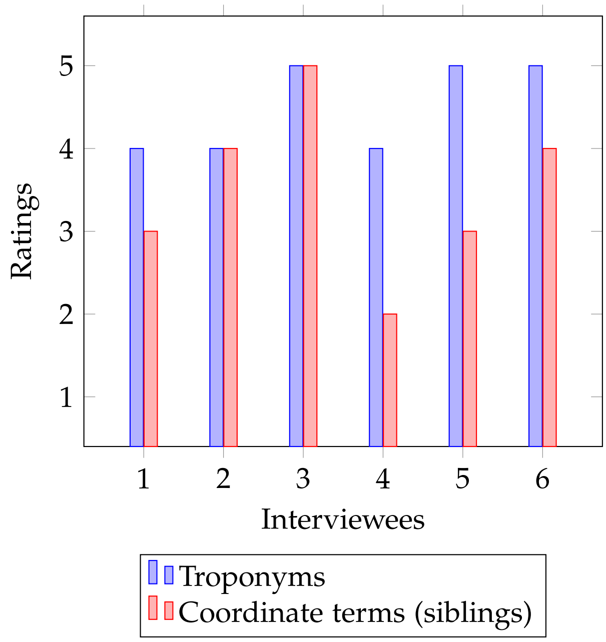


Figure 3.3.: Ratings to the question: Suppose you got a case about discrimination against handicap. How useful would you find: . . . ? (Scale from 1 to 5).

partial or total loss. Therefore, such information would not be helpful.

Table 3.4.: Summary of average ratings for the third situation.

	Average rating	**%**
Meronyms	2.8	47.2%
Holonyms	2.7	44.4%

It was motivating to listen to some of the interviewees mention that integrating this kind of knowledge in a search engine was a great idea and that this information would definitely support legal searches when used in the correct way. Furthermore, some of them mentioned that the crucial aspect in this field is to restrict the results to the context and that this is precisely what many databases are missing nowadays.

On the whole, we can conclude that hyponyms, troponyms, sibling/related terms and derivationally related terms were the relations that the target users find more helpful. If we imagine the ontological tree of a certain word, we can say that hyponyms, troponyms and derivationally related terms are terms that are placed "below" the word,

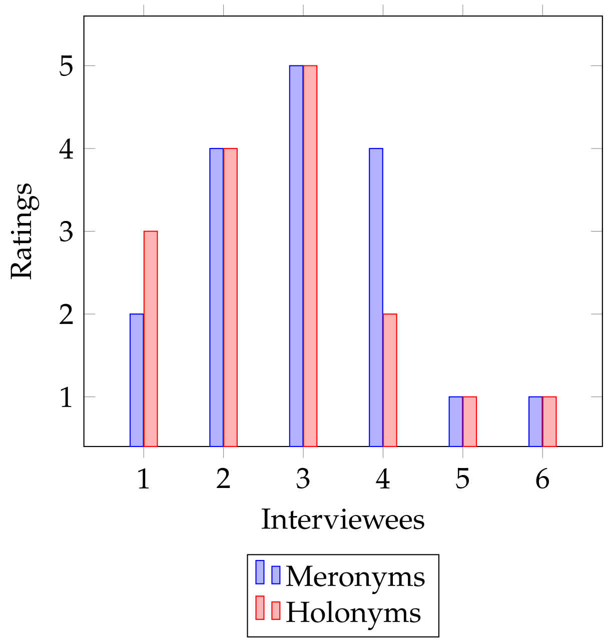


Figure 3.4.: Ratings to the question: If you got a case about a person having an accident during work and as a result the right hand of the worker was seriously injured. How useful would you find a system that suggests: . . . ? (also in a 1 to 5 scale).

while siblings are placed in its same level. Therefore, to build the prototype system, hyponyms were selected to exemplify the integration of the lexical information to the system, as well as hypernyms combined with sibling terms.

The next section describes how the prototype system is designed.

3.5. System design

This section presents the design and architecture for the final system as well as the technology required to built it. Also, the system interface and processes are described. Additionally, this section defines the use cases and presents the data sources used for the database.

3.5.1. Technology used

Here is presented a list of the main technologies used to develop the whole system with a brief description for each of them:

1. **AngularJS** – is a client-side structural open-source framework for dynamic web applications (Google, 2010-2015a). It extends HTML syntax to express application components and supports the dependency injection and the Model-View-Controller design patterns.

2. **Elasticsearch** – provides a distributed restful real-time search server and analytics. It is built on top of Apache Lucene (The Apache Software Foundation, 2011-2012), provides full-text search and works with JSON documents (Elasticsearch BV, 2015a).

3. **Play Framework** – is a web application framework for highly scalable applications in Java and Scala. It is based on a lightweight, stateless and web-friendly architecture (Play, 2014). Moreover, Play follows the MVC (Model-View-Controller) architecture pattern.

4. **GermaNet** – as mentioned before in this thesis (see subsection 2.1.2), it is a lexical semantic net for German developed at the University of Tübingen.

5. **ElasticUI** – is a set of AngularJS directives for elasticsearch (YousefED, 2014). It uses the browser build of elasticsearch for AngularJS and the JavaScript implementation of the elasticsearch DSL (Domain Specific Language) (FullScale Labs, LLC., 2012-2014).

6. **Bootstrap** – is a front-end (HTML, CSS and JS) framework to build responsive applications (Otto & Thornton, 2011-2014). Its components make web development faster and easier.

7. **UI Bootstrap** – is a set of Bootstrap components written in AngularJS (Angular-UI team, 2014). These directives aim to have no dependency in jQuery and they are added as a dependency to the AngularJS module of the application.

8. **JSON** – (JavaScript Object Notation) is a format for lightweight data-interchange (JSON, n.d.). It is completely language independent; however, some familiar conventions to programmers are used.

9. **Git** – is a tool for version control (Git, 2015).

10. **Bitbucket** – is a hosting site for Git projects (Atlassian, 2015).

3.5.2. System architecture

Many considerations were taken into account when building the architecture for the system. At first, the goal was to develop the whole system with AngularJS framework. However, GermaNet API is built on Java and this resulted in complications when combining it with AngularJS.

A second attempt was made to develop the whole system with Play Framework, since Java would not be an issue here. Nevertheless, difficulties aroused from combining Play Framework with Elasticsearch due to compatibility issues and lack of documentation between the Elasticsearch plugin for Play and the Play Framework's latest release.

Consequently, a more appropriate solution resulted in a combination of building two applications, one with AngularJS and one with the Play Framework. This way, GermaNet can be integrated in a Java environment and Elasticsearch integrated with AngularJS without a problem.

RESTful (*Representational State Transfer*) requests are used to connect both of the applications. REST is an architectural style that provides scalability and reuse of the components (Fielding & Taylor, 2002). Also, it attempts to minimize network communication, which makes it a perfect choice to perform the connection between the two applications. Figure 3.5 presents the final architecture.

This architecture supports scalability, since the application involving GermaNet (i.e. GermaNet Wrapper) is completely separated from the main application (i.e. The Golden Retriever). We know that GermaNet is a good lexical database for German and this allows one to prove the concept of this thesis. However, GermaNet has limitations regarding the amount of German legal vocabulary that is included in it and perhaps, in the future, this module could be replaced by a more powerful ontological database focused on legal material that produce better results in this domain. Therefore, the separation of these modules foresees and overcomes these further changes.

It is important to note that with this solution there will be three servers running in parallel so that the application can work: the AngularJS web-server (npm - node package manager (Joyent, Inc., 2015)), the Play Framework server (through Typesafe Activator (Typesafe, Inc., 2015)) and the Elasticsearch server.

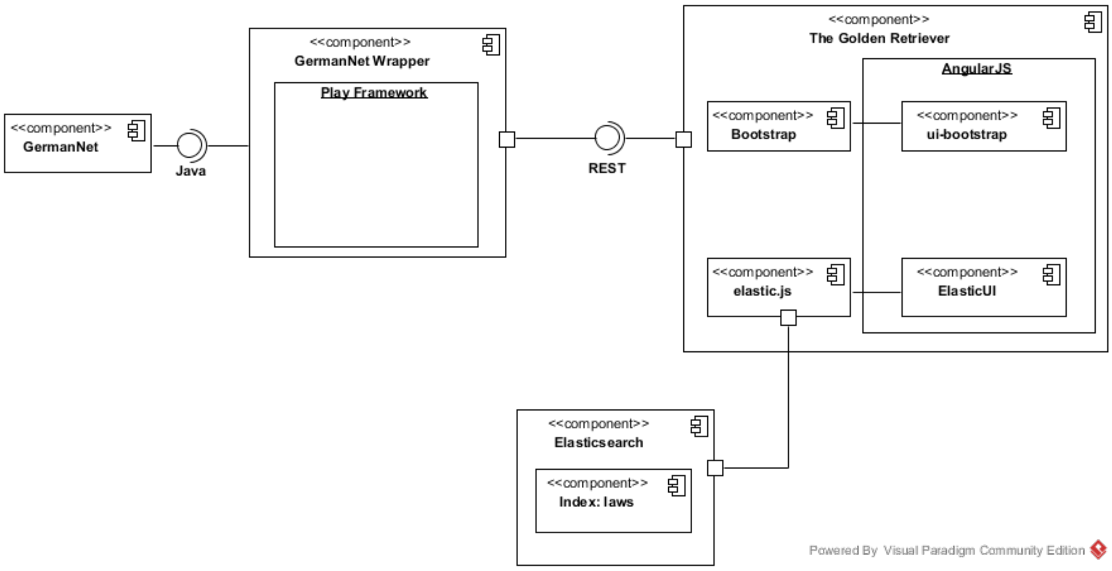


Figure 3.5.: System architecture.

Component description
An overall description of each of the architecture components is presented in the following lines. More details will be discussed in the following chapter.

- **GermaNet** – is the document database that GermaNet API requires to provide the ontological information.
- **GermaNet Wrapper** – is the application that serves as a lexical information retriever. It accesses the GermaNet database and creates the appropriate methods to return the lexical relations (i.e. hyponyms and hypernyms) of a given word. This component is built with Play Framework.
- **The Golden Retriever** – is the main module of the whole system, i.e. the search engine that retrieves the golden results. It is built with AngularJS framework and uses Bootstrap and UI Bootstrap to build the front-end. Furthermore, this module integrates the ElasticUI directives and elastic.js to retrieve information from the Elasticsearch database.
- **Elasticsearch** – is the database server. Here all legal documents used are going to be indexed.

3.5.3. Mockups

The search user interface overview is presented in Figure 3.6. This interface was designed trying to keep all elements as simple as possible. It allows the user to introduce a query (search term(s)) and it shows the list of the search results. On the left-hand side, the search support mechanisms are displayed in two formats: as a breadcrumb and as a list of clickable elements contained in separated boxes categorized by lexical relation.

Furthermore, the result list only shows the text fragment where the input word was found in each hit. An expanding mechanism for the text fragment is also provided. In addition, a paging mechanism is also implemented, as well as displaying the number of hits (*Treffer*).

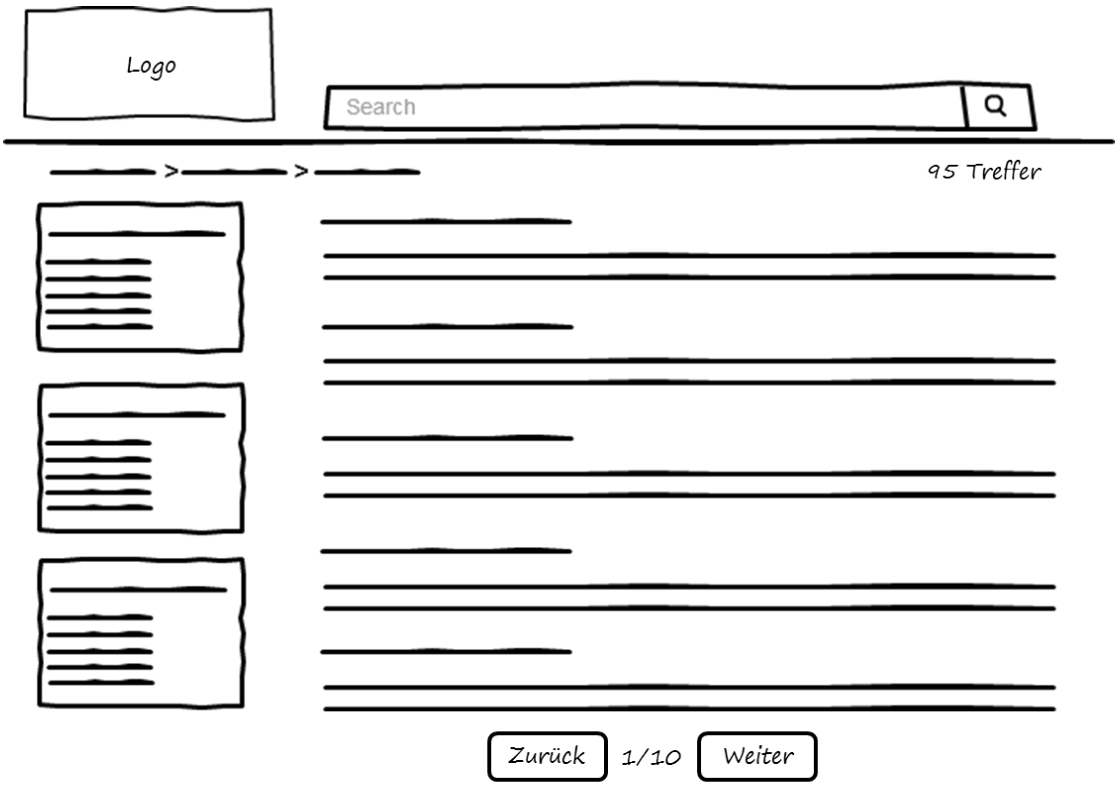


Figure 3.6.: Search user interface mockup.

Figure 3.7 presents an example of the usage of this interface by searching for the word *"Wohl"*(Welfare), where the hyponyms (in this context presented as *Unterbegriffe*) are presented in the container box titled ***Suche eingrenzen*** (Narrow your search).

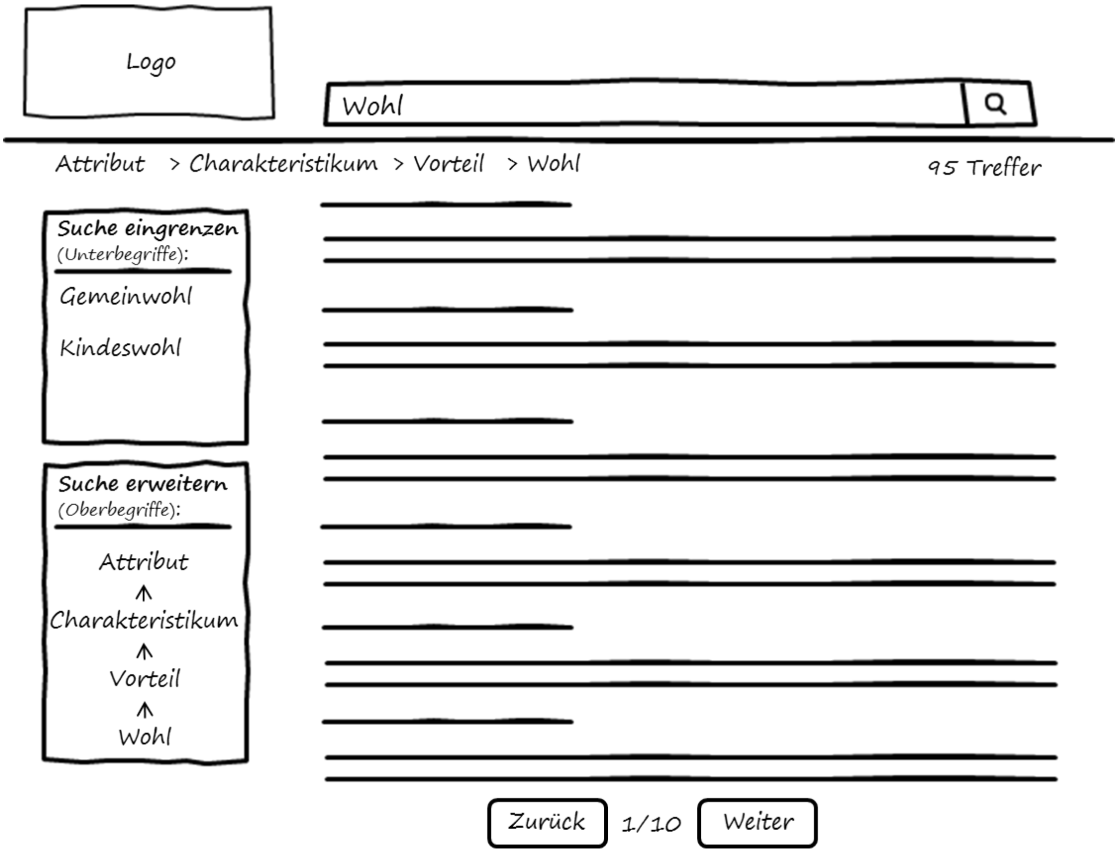


Figure 3.7.: Search user interface mockup example.

The hypernymy lexical relation was more challenging than the hyponymy since it is more complex, in the sense that it is represented by a hierarchy of words. Therefore, hypernyms cannot or should not be presented as a list like the hyponyms, since the relation between the words would get visually lost to the users.

Four proposals to present this relation were analyzed:

1. **Bullet points**
   Presenting the set of hypernyms as bullet items would show the user the relation of belonging of one item to the other. Figure 3.8 presents the idea with the hypernyms of the word *Wohl*. This proposal resembles a table-of-contents view. However, there are words that have an extensive number of hypernyms, thus the width of the container where they would be presented might be too large for a sidebar element.

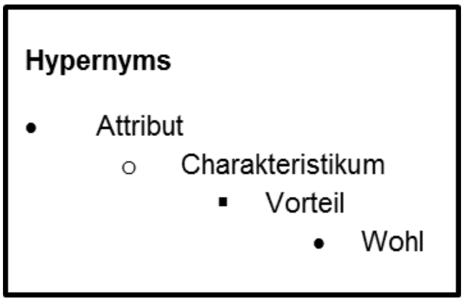


Figure 3.8.: Bullet points proposal for hypernyms.

2. **Colored hypernyms**
   The second proposal was presenting the hypernyms with a degradation of colors getting more intense as they get closer to the input word (i.e. the search term). This would help the user to visualize how strongly related the words are depending on the intensity of the color.
   Nevertheless, the hierarchy relation between the terms might be also confusing since they are presented as a list. Plus, when there are many related hypernyms, the coloring between two consecutive ones might become too similar that it can get difficult to distinguish the colors. *(See Figure 3.9).*

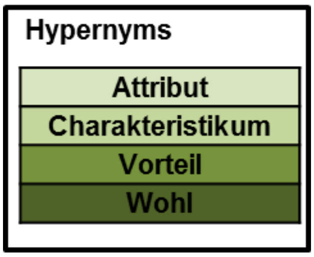


Figure 3.9.: Color degradation proposal for hypernyms.

3. **Tree**
   Presenting the hypernyms in a tree-like structure was the most visually intuitive approach. Arrows can be added between the words and this would show the relation between them explicitly. Therefore, this proposal was selected to be implemented in a container on the sidebar of the interface (***Suche erweitern:*** *(Oberbegriffe)* (Expand your search: (Hypernyms)) in Figure 3.7).

4. **Breadcrumbs**
   This last proposal is presenting the elements a in a breadcumb element. Since

breadcumbs are common patterns in web applications and also known for supporting navigation, it should not be problematic for the users to figure out the relation between the words presented as a navigation trail (i.e. breadcrumb). Consequently, also this proposal was selected to be implemented. This tool is placed on the top of the sidebar and the result list, see Figure 3.7.

3.5.4. Process

Figure 3.10 presents how the basic process of searching by using lexical information in the system will be handled.

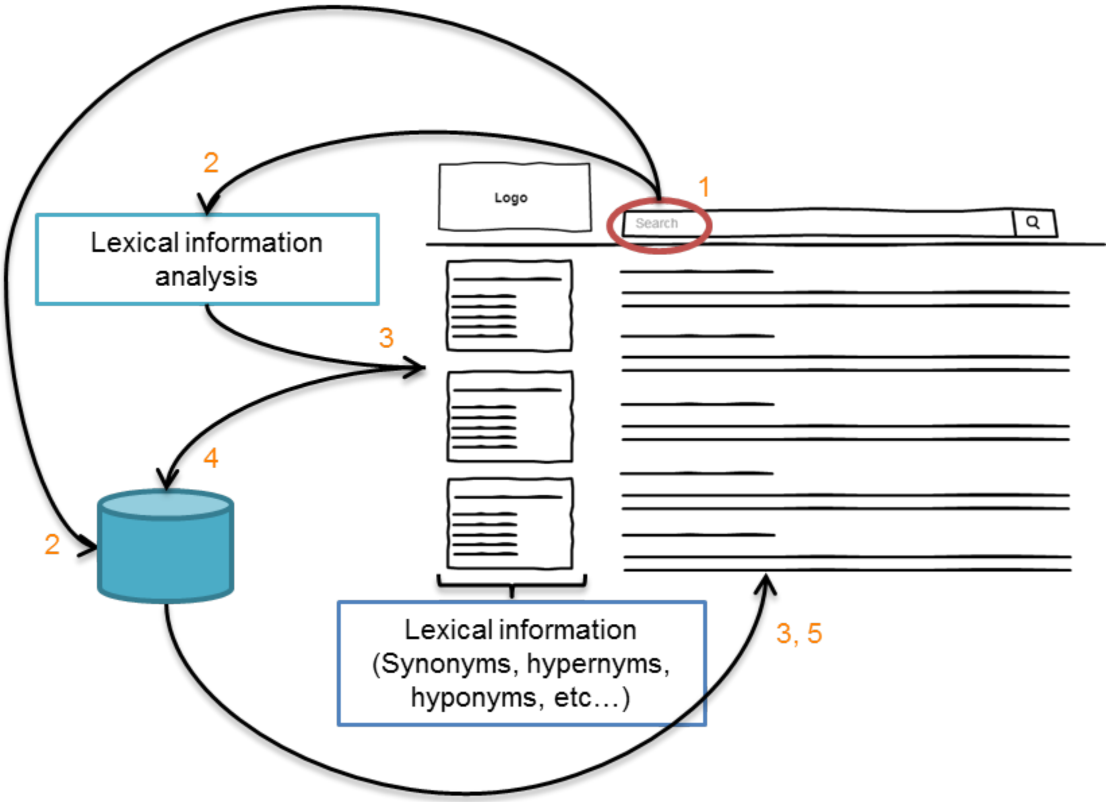


Figure 3.10.: Basic system process diagram.

Steps:

1. The user introduces a word to search for in the input field.

2. The system searches for matches and retrieves the lexical information of the input word.

3. The system shows this lexical information as a search support mechanism and the search results.

4. The user can select terms from the lexical information shown and narrow or expand the query. When a term is selected, the system searches for matches.

5. Results are updated according to the lexical information selected.

Figure 3.11 formalizes this process in a sequence diagram. The instances for this diagram are the three components required to run the application plus the legal-database user.

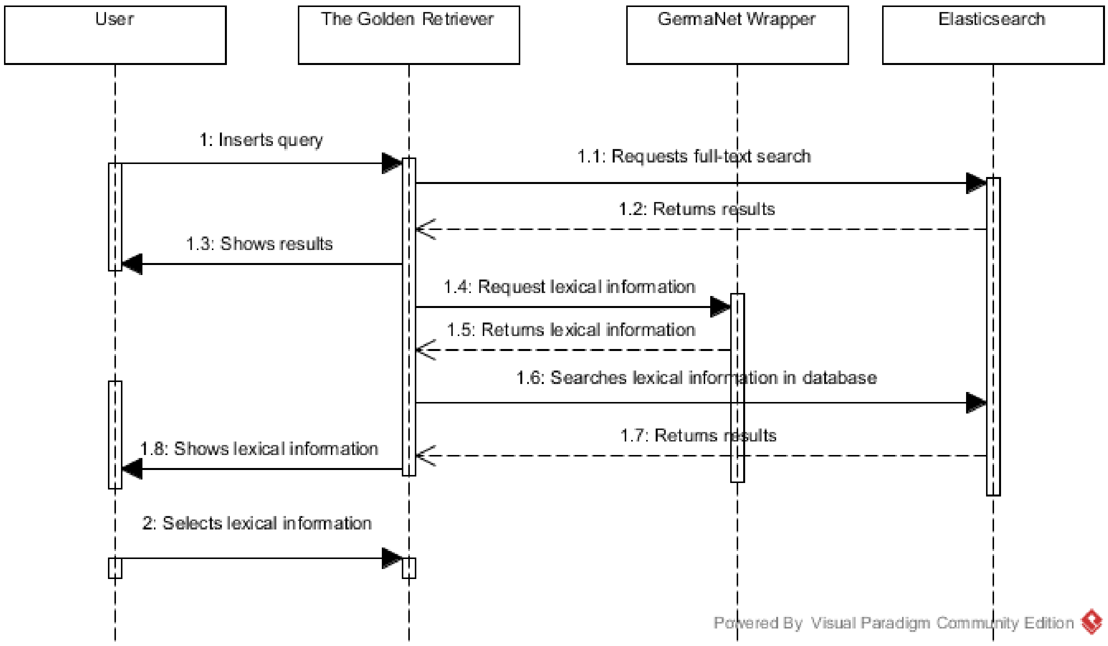


Figure 3.11.: Sequence diagram.

In the sequence diagram we can see that some activities occur simultaneously. This is because the main module (i.e. The Golden Retriever) must request the full-text search results to Elasticsearch while also it requests the lexical information from GermaNet Wrapper and the matches of the lexical information in the database. All this with the purpose of displaying this information and the results at the same time.

After the user performs the last activity, the whole process repeats itself; but this time, the query inserted will be the lexical information that the user selected.

3.5.5. Use cases

After the process has been described, Figure 3.12 summarizes the main activities the target users can perform while interacting with the system. In this context, we only have one type of role to satisfy: the legal-database user.

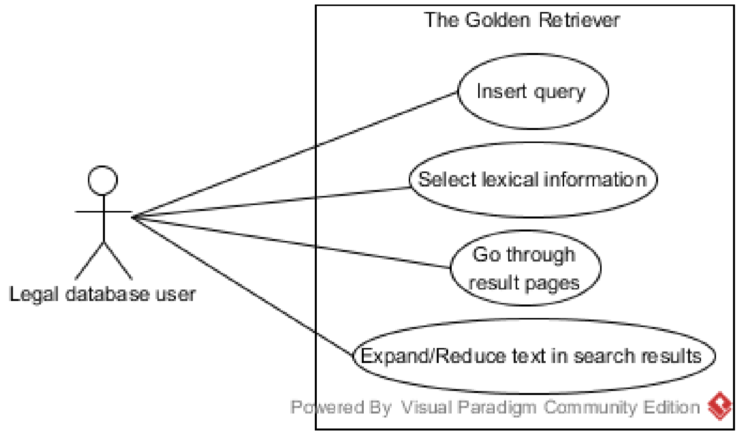


Figure 3.12.: Use cases diagram.

3.5.6. Data sources

A selection of 14 German law documents (Juris, 2013) was made to serve as database of the system:

1. **AGG** – *Allgemeines Gleichbehandlungsgesetz* (General Equal Treatment Act)
2. **ArbZG** – *Arbeitszeitgesetz* (Working hours Act)
3. **BDSG** – *Bundesdatenschutzgesetz* (Federal Data Protection Act)
4. **BGB** – *Bürgerliches Gesetzbuch* (German Civil Code)
5. **GG** – *Grundgesetz für die Bundesrepublik Deutschland* (German Constitution)
6. **GlasVMstrV** – *Glasveredlermeisterverordnung* (Regulations for Glass Refinement Master Craftsmen)
7. **KAGB** – *Kapitalanlagegesetzbuch* (German Investment Code)
8. **KWG** – *Kreditwesengesetz* (Banking Act)
9. **ProdHaftG** – *Produkthaftungsgesetz* (Product Liability Law)

10. **StGB** – *Strafgesetzbuch* (Criminal Code)
11. **TKG** – *Telekommunikationsgesetz* (Telecommunications Act)
12. **TMG** – *Telemediengesetz* (Telemedia Act)
13. **TPG** – *Transplantationsgesetz* (German Transplantation Law)
14. **WaffG** – *Waffengesetz* (Weapons Law)

Each of these documents is built in XML format. Figure 3.13 shows the attributes that compose them and in Figure 3.14 we can see a diagram of how these attributes are interconnected. Please note that not all the attributes are utilized in the system; this will be discussed in the following chapter.

- Document
  - Build date (*buildDate*)
  - Document number (*dokNr*)
  - Law (*Norm*)
    - Build date (*buildDate*)
    - Document number (*dokNr*)
    - Metadata (*Metadaten*)
      - Judicial abbreviation of the law (*jurAbk*)
      - Official abbreviation of the law (*amtAbk*)
      - Creation date (*ausfertigungsDatum*)
      - Reference (*Fundstelle*)
        - Reference type (*typ*)
        - Reference document (*periodikum*)
        - Citation information (*zitstelle*)
      - Short title (*kurzUe*)
      - Long title (*langUe*)
      - Structuring unit (*gliederungseinheit*)
      - Name (*enBEZ*)
      - Title (*titel*)
      - Version information (*Standangabe*)
        - Checked
        - Edition type (*standTyp*)
        - Edition comments (*standKommentar*)
    - Textdata (*Textdaten*)
      - Content (*Inhalt*)
      - Footnotes

Figure 3.13.: German law document attributes.

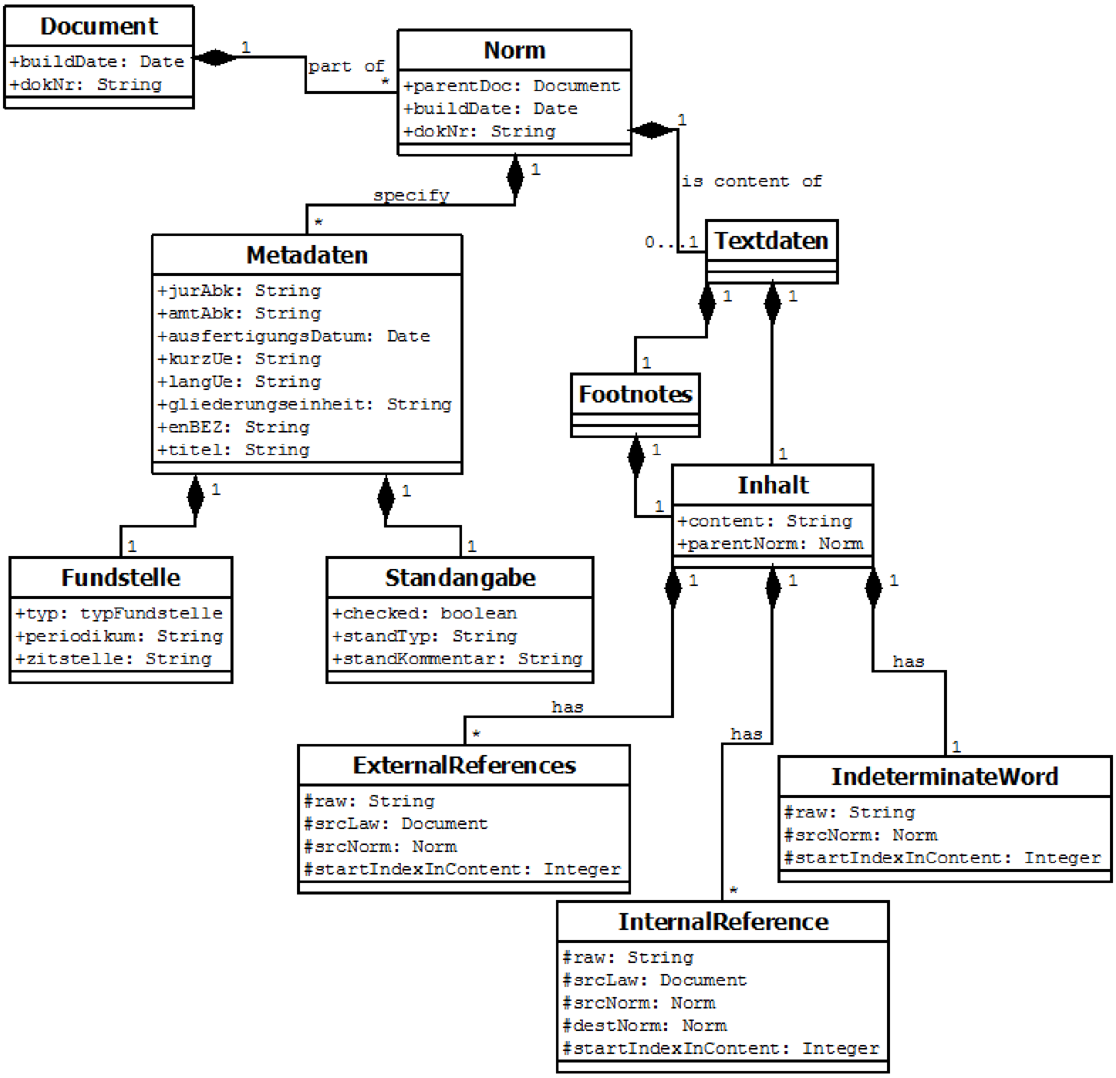


Figure 3.14.: German law class diagram.

4. Implementation

This section presents a detailed and technical description of the main two system components (i.e. GermaNet Wrapper and The Golden Retriever) and how they integrate the other modules (i.e. GermaNet and Elasticsearch) into the system. Furthermore, each implemented feature is defined, including the search support mechanisms integrated with the selected lexical relations to implement.

4.1. System modules

4.1.1. GermaNet Wrapper

This module integrates GermaNet version 9.0 and Play framework version 2.3 provides the server to run it. It lacks of a proper graphical user interface, since the users would not be interacting directly with it.

Regarding its functionality, GermaNet Wrapper returns all the lexical information needed, by requesting it to the GermaNet database through the following two methods:

1. `controllers.Application.getHyponyms(String word)` – This method receives a word as an input. Then, it retrieves its lexical unit (word sense, e.g. if the word is a noun or a verb) by calling the GermaNet method `getLexUnits (String orthForm)` and retrieving only the first word sense of the list that this method returns. Now, from this lexical unit we can call the `getSynset(). getRelatedSynsets(ConRel type)` method and set `ConRel.has_hyponym` as the relation needed (i.e. type). This operation will retrieve first the synset (i.e. the set of all (almost) synonym words) of the lexical unit and with this synset we can retrieve a list of all the related hyponym synsets.
   The synset list is iterated to extract all the words that compose each of the synsets. These words will be returned in JSON format to the requester (i.e. The Golden Retriever).
   To exemplify this process, we can take the word *Schaden* (damage). First, we retrieve an orthographic form of the different word senses. These senses can be, for example, *Schaden* as a noun (damage) or *Schaden* as a verb (harm). The system will only take the first sense in the list that GermaNet returns, i.e. damage.

Now, we get all the related hyponym synsets for this word, which are:
(Each bullet point represents a related synset)

- *Schande* | *Schmach* | *Schimpf* (disgrace | shame | insult)
- *Sachschaden* | *Schadensfall* (property damage | claim, case of damage)
- *Brandschaden* (fire damage)
- *Schändung* | *Entweihung* (violation | profanation)
- *Zerstörung* | *Destruktion* (destruction)
- *Vermögensschaden* (property loss)
- Among others...

Finally, a list containing all these elements is the one that is returned to the requester.

2. `controllers.Application.getHypernyms(String word)` – This method receives a word as an input and returns a hypernym list. In the same manner as the method before, it starts by retrieving the lexical unit of the word from GermaNet. Afterwards, an auxiliary method is called, which will recursively fill a list with all the hypernyms of the word.
The recursive auxiliary method gets the hypernyms very similarly to the `getHyponyms` method; however, here the relation specified to retrieve the synsets is the `ConRel.has_hypernym`. Likewise, only the first word sense of the word is retrieved and this is what is added to the hypernyms list to return.
This method needs to be recursive since in GermaNet we can only retrieve the direct hypernym of the word. Therefore, we must climb one step of the word's ontology tree in each iteration and request the next hypernym until the top (i.e. GNROOT). When the step is a synset, only the first word in it is retrieved. This word will represent the synset in the hypernym list.
In the case of the word *Schaden*, firstly, the method retrieves the first word sense of it: *Schaden* as a noun (equally as in the former method). Subsequently, the auxiliary method fills a list with the word's hypernyms. This list will contain the following elements:
    - *negativer Vorfall* (negative incident)
    - *Vorfall* (incident)
    - *Geschehnis* (event)
    - *Ereignis* (event)
    - *Situation* (situation)

- *Zustand* (state)
- *GNROOT* (GermaNet element)

As mentioned in the methods' description, the system will only retrieve the first word sense of the word in question. This is because sometimes words can be very ambiguous, which is a very well-known problem called "Detection of polysemy" or "Word sense disambiguation". This problem refers to words that can be polysemous, i.e. have different meanings. For instance, the word "Table" can refer to furniture but also, to an arrangement of data in columns and rows.

Determining which sense is the correct one for a specific case depends mainly on the context where the word is used. Therefore, retrieving only the first sense listed of the word is a trivial approach to tackle this problem. It might not be the best approach to get the correct word sense, but it is a common solution. Besides, this problem certainly requires a dedicated research, thus, it is out of the scope of this work.

4.1.2. The Golden Retriever

This module produces the main interface of the system where the users can insert queries and go through the search results.

First, we know that Elasticsearch works with JSON documents, nonetheless, the data sources (i.e. the 14 selected law documents) are built in XML format. Therefore, to be able to index these data sources in the database, a small application needed to be developed to parse the files from XML into JSON format. To perform this step, an application developed within the Software Engineering for Business Information Systems chair (i.e. The Law Importer) was reused and extended. This application transforms the XML document elements into Java objects and with those we can produce a JSON object and print it in a file. Also, an identification number was added to each element.

After this, each file can be bulk indexed into Elasticsearch. Each element in the files was indexed as a "law" item in an index named "laws". Bulk API allows us to index each document in a single call which makes this process faster (Elasticsearch BV, 2015b).

Once the documents were indexed into the database, the main interface of the system was built with AngularJS to present the data to the user. ElasticUI directives made possible a smooth integration between Elasticsearch and AngularJS.

Let us have a close look to each of the features in the following section.

4.2. System features

4.2.1. Search

In Figure 4.1 we can see the main interface of the system. All the data sources are listed in alphabetical order and the total number of elements (*Treffer*) is displayed. The interface also shows an input field where users can introduce their queries.

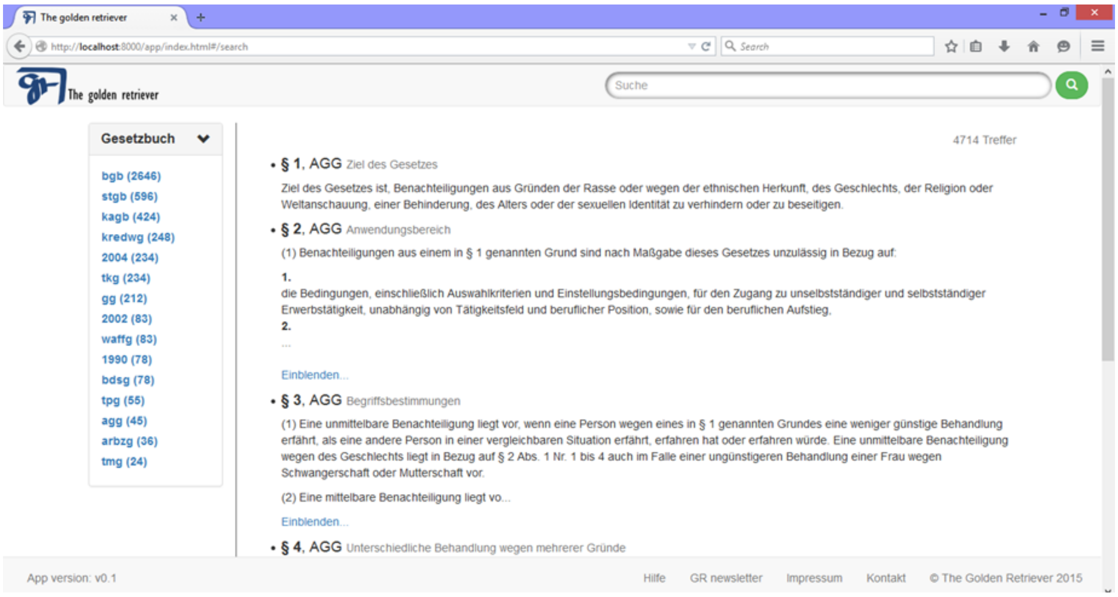


Figure 4.1.: The Golden Retriever main interface.

The search is performed in the following attributes of the data sources: `textdaten.Inhalt` (textdata.content), `metadaten.titel` (metadata.title) and `metadaten.jurAbk` (Judicial abbreviation of the law in metadata).

In addition, faceted navigation is included in the system on the left-hand side. Here, the elements are categorized according to the legal document (*Gesetzbuch*) they belong to; plus, it is shown how many matches of each of the categories are within the dataset. This feature is implemented with ElasticUI's euiSingleselect widget which makes use of Elasticsearch's aggregations module (which gives the faceting functionality). Furthermore, it is build with UI Bootstrap's accordion directive which allows the user to collapse or expand the list's container.

What is more, each of the elements in the result list is presented with the number of the law (e.g. §1 in the first listed element), the legal document it belongs to (e.g. AGG) and the title of the law (e.g. *Ziel des Gesetzes* (Target of the law)) plus a fragment of its contents. Each fragment can be expanded (*Einblenden*) when it is larger than the space

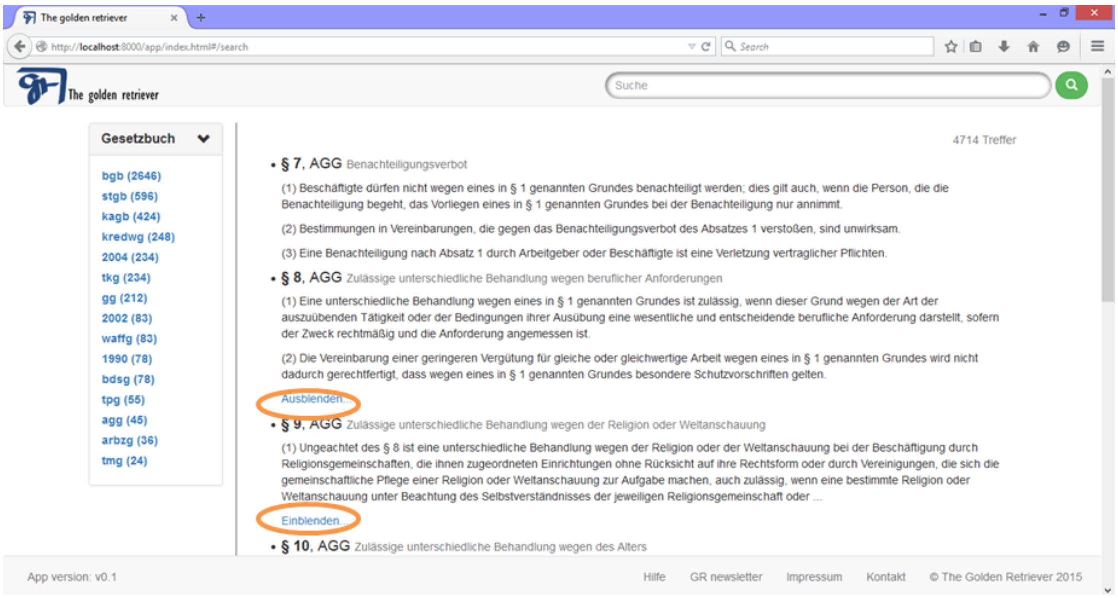


Figure 4.2.: Expand/Reduce text feature.

provided and, naturally collapsed (*Ausblenden*) after it has been expanded. Figure 4.2 shows this functionality.

Furthermore, highlighting the queried terms in the law's content is implemented through ElasticUI's euiHighlight component (Figure 4.3) and the paging mechanism through ElasticUI's euiSimplePaging widget (Figure 4.4).

4.2.2. Search support mechanism for hyponyms

Hyponyms of the queried word are implemented in the interface on the left-hand side as a list in an accordion container titled ***Suche Eingrenzen:*** *(Unterbegriffe)* (Narrow your search: (Hyponyms)) (Figure 4.5). It is important to mention that by the time being, the system only produces lexical information for the first word typed in the input field.

A factory service (Google, 2010-2015b) is created in The Golden Retriever to be able to extract the hyponyms list from GermaNet. This service represents a RESTful client that will make the requests to GermaNet Wrapper with the word the user queried.

Each hyponym is presented together with the number of that hits it has within the database; however, only the terms that actually have matches within the data sources are displayed in the interface. That is, after the hyponym list has been retrieved, a search is performed for each of the elements within the whole search space with the integration of the Elasticsearch Query DSL javascript implementation, i.e. elastic.js. If

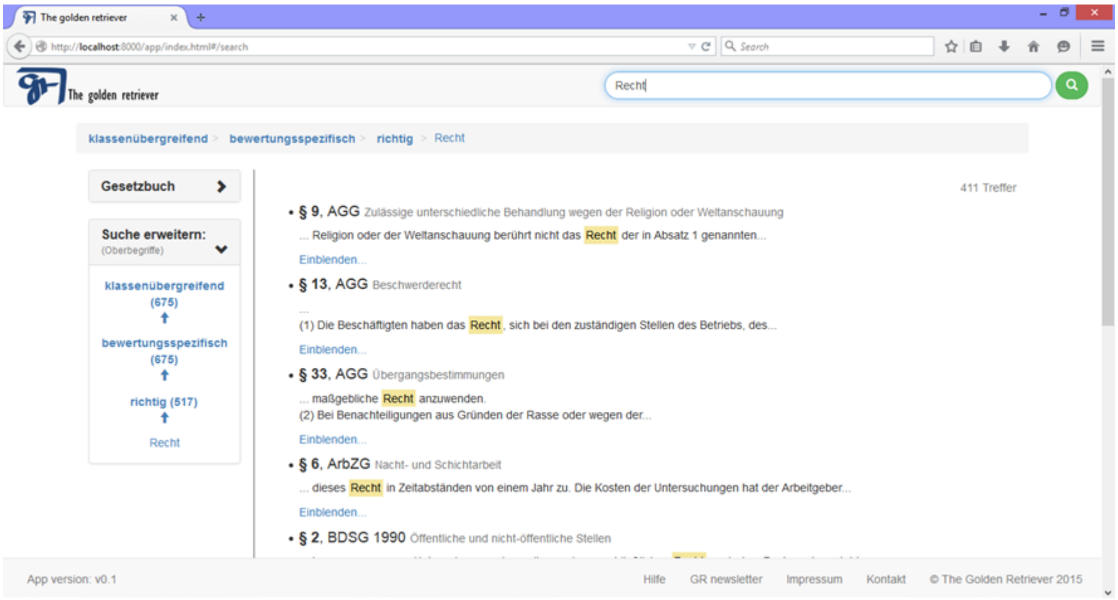


Figure 4.3.: Highlighted query terms within the results.

the element does not match in any of the documents, this element will not be presented to the user.

It may seem that this search support mechanism can be categorized in the *Faceted Navigation* division; however, the search performed to retrieve the number of matches of hyponyms within the database is performed through the whole search space. Therefore, this mechanism cannot be called a proper facet since the query results are not the ones that are categorized, but the matches from the whole dataset.

Consequently, this search support mechanism, for this particular lexical relation, fits better the *Query refinement suggestions* category, which is part of the *Automated term suggestions/query expansion* tools. These suggestions help the user to narrow the search and/or to add context to it; plus, the query is replaced by the new selected hyponym and issued again.

4.2.3. Search support mechanisms for hypernyms

Hypernyms are also extracted by querying GermaNet Wrapper through another method in the factory service.

As mentioned before, this relation is presented in two different ways:

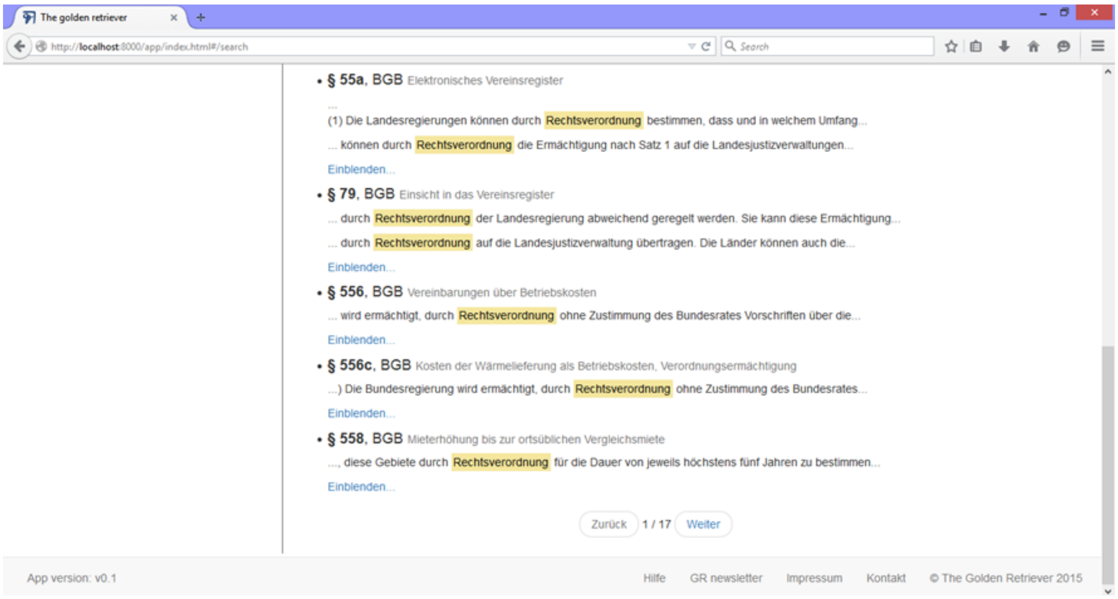


Figure 4.4.: Paging mechanism.

1. Hypernym tree

The hypernym tree, similar to the hyponyms tool, is presented in an accordion container. Nevertheless, this container encompasses here a hierarchy tree of the word's hypernyms. This tree starts with the input word and ends in the last hypernym before GNROOT. GNROOT is not included since it is a GermaNet element and it is certainly not expected to be part of any of the documents' contents.

Together with each hypernym, it is displayed a number of hits. These hits however, are not only the number of matches of the hypernym within the dataset, but the number of matches of each hypernym plus its hyponyms, i.e. the siblings of the word in question. To put it simple, let us look at the following example:

In Figure 4.6 a search is performed for the word *Telekommunikation* (Telecommunication). On the left-hand side, a container for the word's hypernyms is presented (***Suche erweitern:*** *(Oberbegriffe)* – Expand your search: (Hypernyms)). Within the results, the only word that is highlighted is the input word.

In the hypernym tree, the term *Telekommunikation* is added at the beginning to indicate its root. The next term (*Kommunikation* (Communication)) is presented next to the number *67*. This number represents the number of matches of the term itself plus the hyponym terms of *Kommunikation*, for example: *elektronische Kommunikation* (Electronic communication), ***Telekommunikation*** (Telecommunication), *Telefonverkehr* (Telephone traffic), *Briefwechsel* (Correspondence), *Briefverkehr* (Correspondence), among others. All

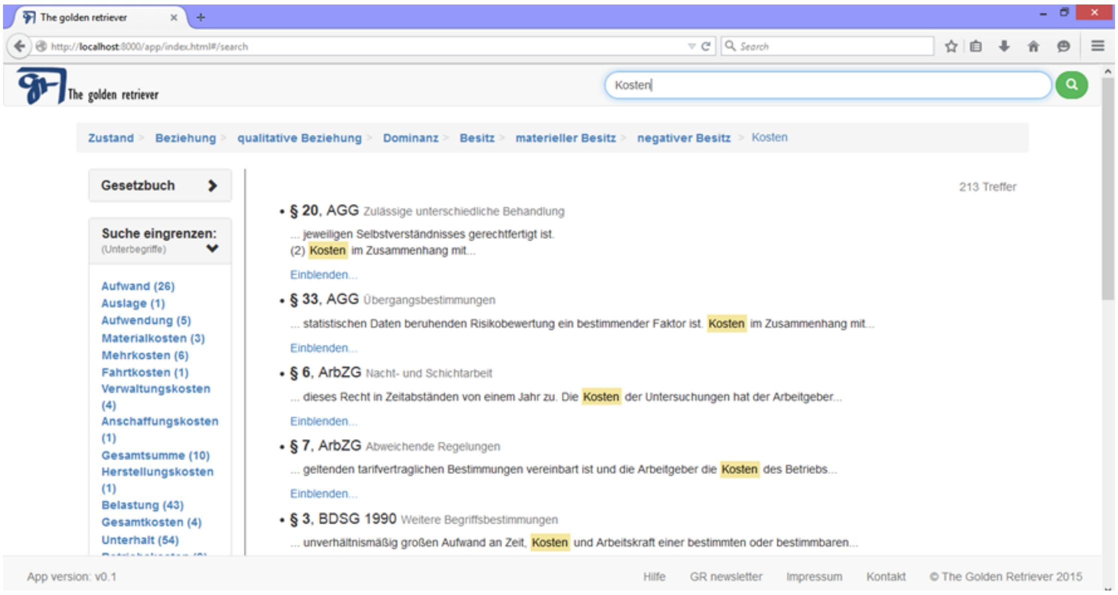


Figure 4.5.: Hyponym search support mechanism.

these terms are located at the same ontology level that the queried word (i.e. siblings). It is noteworthy that the element below *Kommunikation* in the tree hierarchy is, of course, also added to the list.

The following step in the tree, i.e. *Beziehung* (Relationship) shows the number of matches of the term itself, plus its hyponyms and the hyponyms of *Kommunikation*. These last hyponyms are added since *Kommunikation* is part of the path until the searched term (i.e. *Telekommunikation*) is reached. However, the hyponyms of the other hyponym terms of *Kommunikation* are not added. This pattern will be followed by the next tree elements and evidently, the number of hits can only be equal or greater to the last one as one climb up in the tree.

To visualize better this relation, let us look at Figure 4.7. Here one element is contained within the other. In regard to the previous example, Z would represent the word *Beziehung* (Relation). Y would represent all the hyponyms of *Beziehung*, which include the word *Kommunikation*. Finally, X would represent all the hyponyms of *Kommunikation*, including *Telekommunikation*.

Since this might get confusing for the users, a popover was added to each of the tree elements in order to show to the user what exactly the number of hits represents (Figure 4.8) . This way the user can see all the terms that influence this match number, i.e. the hyponyms of the word.

Furthermore, all these elements in the tree except from the input word are presented

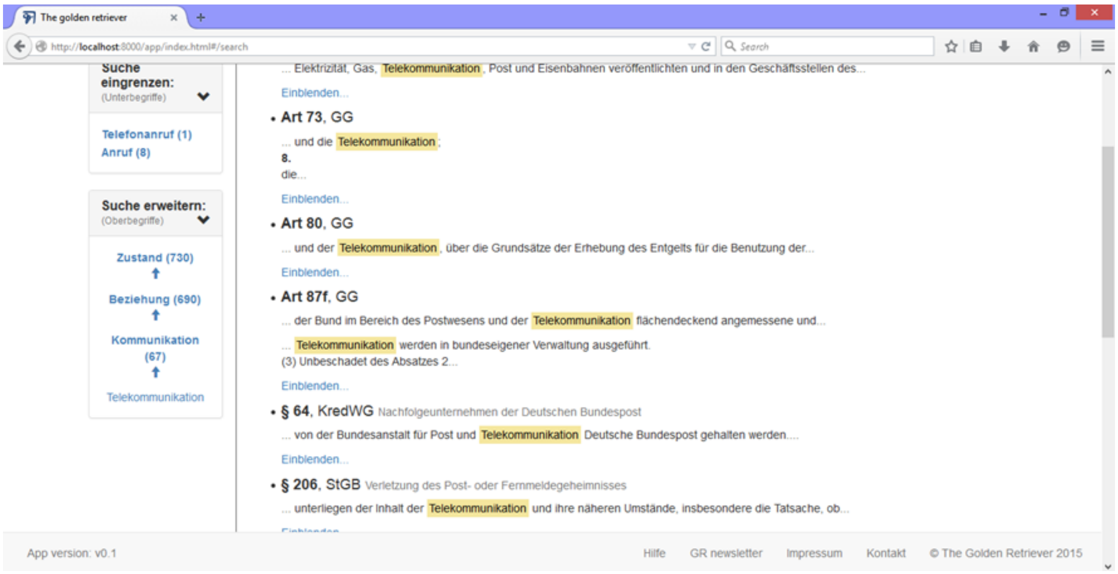


Figure 4.6.: Hypernym search support mechanism.

as clickable elements. When clicked, the element plus its hyponyms will expand the query and the matches will be displayed.

When the query is expanded, the results will refer also to the hyponyms' hits of the selected hypernym, thus another popover is added to the input field to make clear what was searched for (*See Figure 4.9*).

2. Breadcrumb

Another visual tool is added for the hypernymy relation to reinforce the visual recognition of the hierarchy tree. The breadcumb can be found right above the sidebar

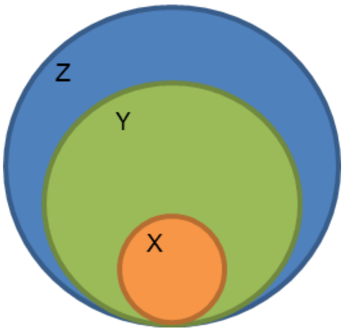


Figure 4.7.: Graphical representation of hypernymy.

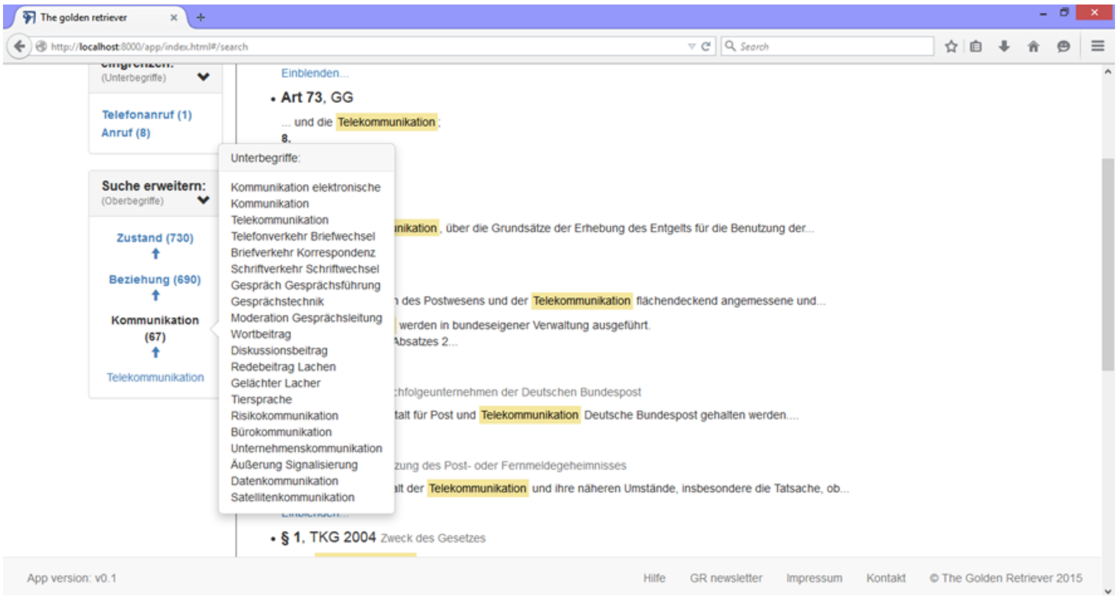


Figure 4.8.: Popover in hypernyms.

in Figure 4.9 . Also, a popover is implemented to show the hyponyms of the term (*Figure 4.10*).

Elements in this search support mechanism are clickable as well, except from the search term. Likewise the hypernym tree, when a hypernym in the breadcrumb is selected, the query will be expanded by the term itself plus its hyponyms and results will be retrieved accordingly.

Taking everything into account, both the hypernym tree and the breadcrumb can be *Table-of-contents views* tools regarding the way they are presented. But they fit better the *Automated term suggestions/Query expansion tools* category, since the query is replaced when an item is selected and elements are alternative suggestions and presented as hyperlinks.

Furthermore, the hypernym tree cannot be categorized either as a proper facet; since, like in the hyponyms mechanism, the hits shown are matches within the whole search space and not only within the result list.

Finally, it is important to note that the hyponym mechanism, as well as the hypernym ones, are query reformulation and automated term suggestions/query expansion tools that help the users to improve their results according to what they need.

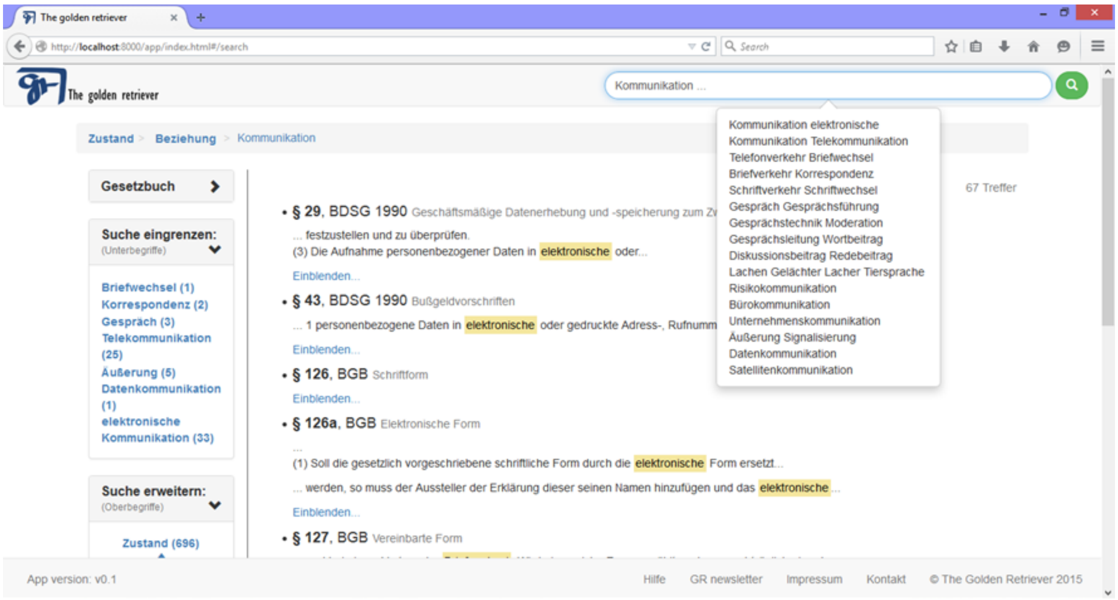


Figure 4.9.: Popover in input field.

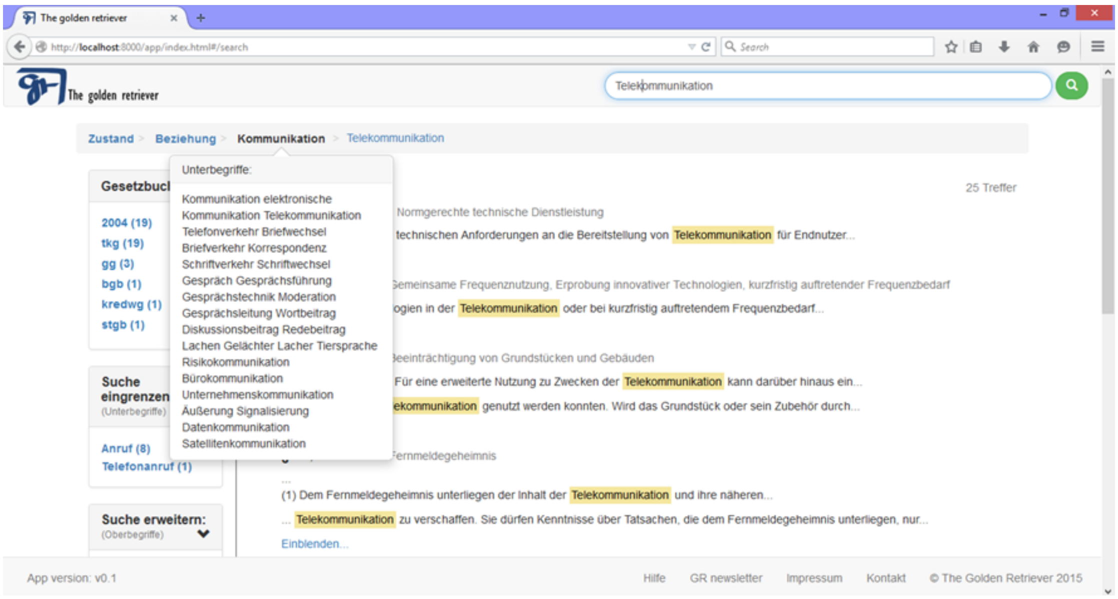


Figure 4.10.: Popover in breadcrumb.

Part III.

Evaluation and conclusions

5. Evaluation

5.1. Limitations

The following limitations are faced by The Golden Retriever system regarding its implementation. They are deduced after analysis made or known from literature:

- **Word sense disambiguation** is one of the most important limitations for this work. As discussed before in subsection 4.1.1, this problem limits the system from extracting the precise word sense from the lexical database of the introduced search term by the user.
- The **search context** is also a challenge for this research, since context is highly relevant for searches in the legal domain. By the time being, the final system does not support context identification. Only a trivial method to retrieve the word sense is implemented. However, this thesis remarks the importance of it and considers it as part of the future work.
- **GermaNet** represents a limitation as well. The Golden Retriever depends on the powerfulness of this database at the moment, regarding the lexical content. We know that GermaNet contains an extensive amount of German vocabulary; nevertheless, it is not focused on the legal domain.

5.2. Empirical results from an expert interview

An expert interview was conducted to evaluate the system. The main interest was to know whether the system is clear for the users and to find out whether the interviewee thinks that the implemented search support mechanisms can actually help a legal database user. Results were favorable.

On the one hand, the user interface was overall rated as very clear and clean. Moreover, the search support mechanisms presented seemed not confusing to navigate through the lexical terms and to display useful information for the search.

On the other hand, some improvements were suggested after selecting a hypernym in the system to expand the query (situation shown in Figure 4.9), since this view was a somewhat complex. For this situation, the interviewee mentioned that adding another popover to each highlighted term within the results would emphasize what exactly the relation of the term to the one that is in the input field is. In this case, users need to know clearly why the system is searching for each of the highlighted terms.

Further suggestions included allowing the user to select or deselect the lexical relations (either in the hyponym or hypernym containers) according to the search they are performing, as relevant results depend on the context. This would upgrade the system to allow the users to have personal settings which would be an added value and would make the search process customized for each single user and case.

Furthermore, the expert mentioned that it is important to include a reference to the lexical database that is being used (i.e. GermaNet); as it is relevant for legal practitioners to know where all this lexical information is coming from.

One more suggestion was to allow the users to hide or display the breadcrumb mechanism, since it contains the same related terms as in the hypernyms container which makes it redundant for the user. However, some users might find the breadcrumbs more comfortable than the container; therefore, the system should allow optional customization.

The popover to see all the hyponyms related to the hypernym term (*See Figure 4.8*) resulted not so favored as it tends to get overpopulated with all the hyponyms and the visual image that it gives is confusing.

To summarize, the advantages that the system offers are:

- **Lexical information presented as query reformulation mechanisms** provides support for a more effective search
- More than one lexical relation (**hyponyms, hypernyms and sibling terms**) is implemented in the system, offering more options to the user.
- **Hyponyms** are presented to the users and they can narrow their searches.
- **Hypernyms and sibling terms** are presented in two different ways as query expansion tools.
- **Clear and clean interface** that allows easy interaction with the lexical information.
- **High potential** of ontologies integrated in legal domain searches.

On the other hand, there are still some areas of improvement and open issues according to this interview:

- Presentation of the **hyponyms' popover** in the hypernym mechanisms needs some improvement to visualize better its contents.

- Adding **reference to the lexical database** used to retrieve the hyponyms and hypernyms.
- Adding an **explanatory mechanism** to each of the highlighted words in the result list to make clearer why the system matched these items when the search is performed from selection of lexical information.
- **Personalization** might improve the users' experience and support them better.
- **Search context** is essential to legal practitioners.

Further improvements and outlook of this research is discussed in the following section.

On the whole, the overall impression that the interviewee got of the system was that it is well-liked and straightforward but some enhancements can upgrade the interface to make it more useful.

6. Conclusions

6.1. Answers to research questions

The following lines report the findings towards the research questions formulated in section 1.3:

- ***How can search quality be improved by lexical information for a legal database?***
  Search quality is improved by lexical information since legal database users would be able to select lexical relations to reformulate their queries. This provides them with new ideas to search and to find out faster what they are looking for.

- ***What mechanisms and methods are common in legal databases?***
  In section 3.3, an analysis was performed by comparing five existent legal databases. Also in the same section, Table 3.1 shows that the following mechanisms are often observed in these specific-domain databases:
  - Autocomplete
  - Parametric search/scoped search
  - Categories for sorting and filtering search results (Relevance criteria)
  - Faceted navigation (Origin/type of publication and Field)
  - Highlight query terms
  - Displaying number of matches

- ***Which search mechanisms and methods can be enhanced by lexical information and how?***
  Also section 3.3 summarizes which search support mechanisms may be possible to be integrated with lexical knowledge:
  - **Autocomplete** – Complete words derived from the ontology tree of a term can be suggested, however, this search support mechanism resulted to be not so relevant to legal practitioners according to the needs assessment performed. Therefore, the resulted system does not implement this mechanism.
  - **Automated term suggestions/query expansion** – The hypernymy and sibling relations can be integrated for this particular search support mechanism.

Hypernyms were implemented in the system as suggested terms and the conjunction of these with the sibling terms of the queried word can be selected to reformulate and expand the search.

– **Query refinement suggestions** – Hyponyms resulted as good candidates for this mechanism. Therefore, the final system integrates this lexical relation. With hyponyms, the search can be narrowed or refined since a hyponym goes a level down in the ontology tree, i.e. the query becomes more specific. For some terms, hyponyms add context to the query. For example, for the term *Verbrechen* (Crime), its hyponym *Steuerdelikt* (tax offence) can narrow the search into the tax domain.

– **Categories for navigating and narrowing** – With these categories one can reduce the results to a specific category group. This was not integrated in the developed system due to the fact that here searches to find hyponyms and hypernyms and fill the search support mechanisms are done through the whole search space and not only through the results. Thus, the hyponyms and hypernyms displayed are not categories for the result list.
However, these categories seem promising when one selects a hypernym from the provided list in the system and wants to narrow or filter the results to a specific hyponym since the system would also search for the hypernym's hyponyms.

– **Organizing search results via table-of-contents views** – Hypernyms are presented as a table-of-content view. However, both tools integrated in the system for hypernyms do not belong to this category as they do not categorize the result list of a search.

– **Faceted navigation** – This mechanism could also benefit from the hyponymy relation to categorize the result list. Like the mechanism before, this was not integrated with lexical information due to the fact that the system only searches for exact matches to the query term. Consequently, the result list cannot be properly categorized into the hyponyms.
Nonetheless, this mechanism is presented in the system to categorize the result list in their correspondent law documents (sidebar in Figure 4.1).

• ***How can a implementation for a support search mechanism integrated with lexical knowledge look like?***
As shown in chapter 4, one search support mechanism for the hyponymy relation was provided as well as two for the hypernymy relation. Both these relations are implemented as query reformulation tools.
In regard to the evaluation, it can be said that legal database users could benefit

from these tools to perform their searches in a more efficient way as the implemented mechanisms provide them with several query-reformulation options.

6.2. Future work

Many improvements could be added to the system developed within this thesis, this section summarizes some relevant enhancements and further development considered.

First of all, The Golden Retriever only presents the lexical information related to the first word typed in the input field. This functionality could be expanded, for example, by adding a mechanism that identifies which is the most relevant word (if the user types more than one) in the complete query. Then, from this word, one could retrieve the lexical information.

Furthermore, the GermaNet database could be improved or replaced by a more powerful database in the legal domain. This would be with the purpose of enhancing the lexical knowledge presented to the user and adding legal context to the search.

Complementary, more than one lexical database could be integrated into the architecture. This way, implementing a mechanism that identifies context from the query might be very useful to spot from which domain the lexical knowledge should be retrieved. And depending on the context, the domain-specific lexical database could be queried. For instance, when one needs to search for a biological term, the system could switch to a biological-domain lexical database.

It would be also interesting exploring the integration of the lexical knowledge in the background of the search. To put it differently, when a user issues a query, the searching also looks for matches to the hyponyms of the queried term and the system could integrate faceted navigation by categorizing results by hyponyms as discussed previously in section 6.1.

Adding lexical relations as part of the parametric search might also provide a good tool for legal database users to perform a more personalized search. Also, as mentioned in section 5.2, personalization can go further by allowing the users to hide/display the breadcrumb tool or by allowing them to select/deselect lexical elements in the provided mechanisms.

Moreover, new ways of presenting the hyponymy relation in the hypernyms list may be found. Currently a popover is displayed in the system to show all the hyponyms related to the hypernym. Nonetheless, the hyponym list can become really large causing a substantially wide popover that might get difficult to read or to be confusing for the users.

6.3. Summary

This thesis presents an approach to integrate lexical knowledge into a German legal database and a system is developed which implements this knowledge in search support mechanisms. The main goal of this work is to determine whether this kind of information actually helps legal practitioners during their searches or not.

Results are only empirical at this point as it is hard to determine how the recall and precision should be measured for this system. However, this work reveals that law practitioners are truly interested in this kind of information for search support and as the system is more enhanced, the more benefit it can bring to the users.

Moreover, this study differentiates from some of the related work presented in section 2.4 since the lexical information here is integrated on the foreground of the system, i.e. users are able to interact with it. As for some other related research, this thesis stands out in the domain since it is done only for the legal area and for the German language.

More importantly, similar to Schweighofer and Geist (2007) investigation, the main improvement/contribution of this work lies on all the new possibilities that the user gets to search, which broads the information to choose and the information coverage is larger. Furthermore, this thesis offers another approach to the proposed relevance feedback of the same authors, to solve the issue that "users may not be able to find a proper search term for the knowledge base as common language may use a different term."

To conclude, many approaches can still be further developed for this particular area. Nonetheless, this research supports and leads to encourage the integration of lexical information with search systems and specifically within the legal domain.

A. Legal database comparison

The following are screenshots from the selected databases where the search support mechanisms can be identified.

A.1. JURION

Figure A.1.: Autocomplete feature. (Wolters Kluwer, 2014a)

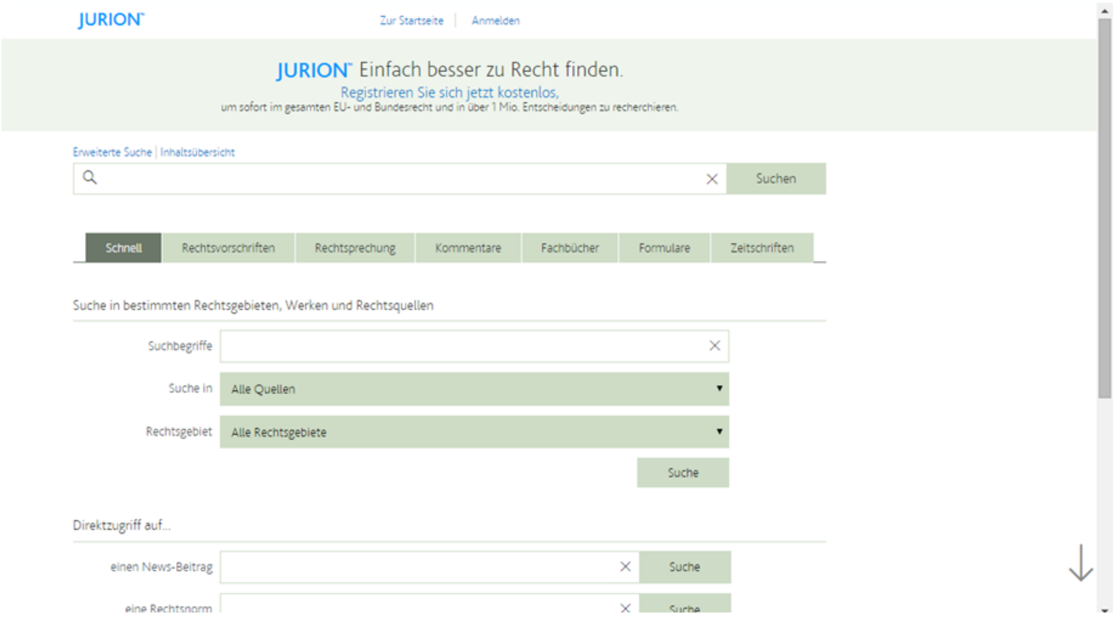


Figure A.2.: Parametric search. (Wolters Kluwer, 2014b)

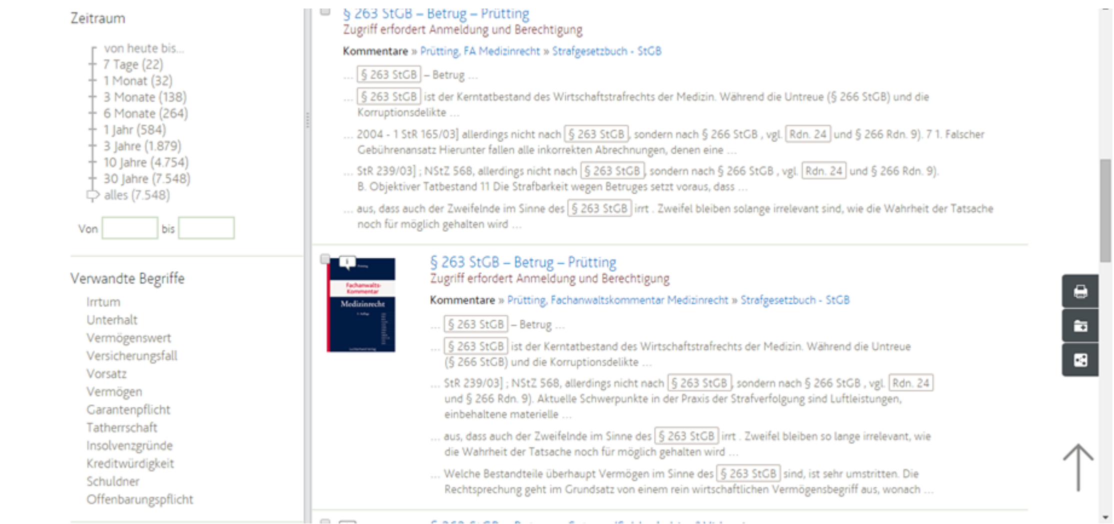


Figure A.3.: Automated term suggestions (Related concepts (*Verwandte Begriffe*)), Highlight query terms and Faceted navigation (Timeline (*Zeitraum*)). (Wolters Kluwer, 2014a)

Figure A.4.: Query refinement suggestions. (Wolters Kluwer, 2014b)

Figure A.5.: Categories for sorting and filtering search results (Relevance criteria (*Nach Relevanz*)), Faceted navigation (Origin/type of publication (*Herkunft*), Number of matches (*Trefferanzahl*) and Letter size change. (Wolters Kluwer, 2014a)

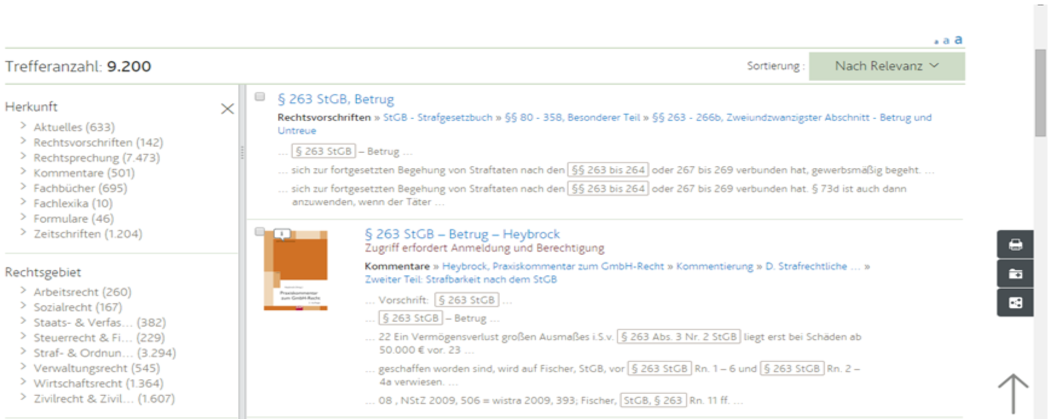


Figure A.6.: Faceted navigation (Field (*Rechtsgebiet*)). (Wolters Kluwer, 2014a)

A.2. beck-online

Figure A.7.: Autocomplete in beck-online. (Verlag C.H.BECK oHG, 2014b)

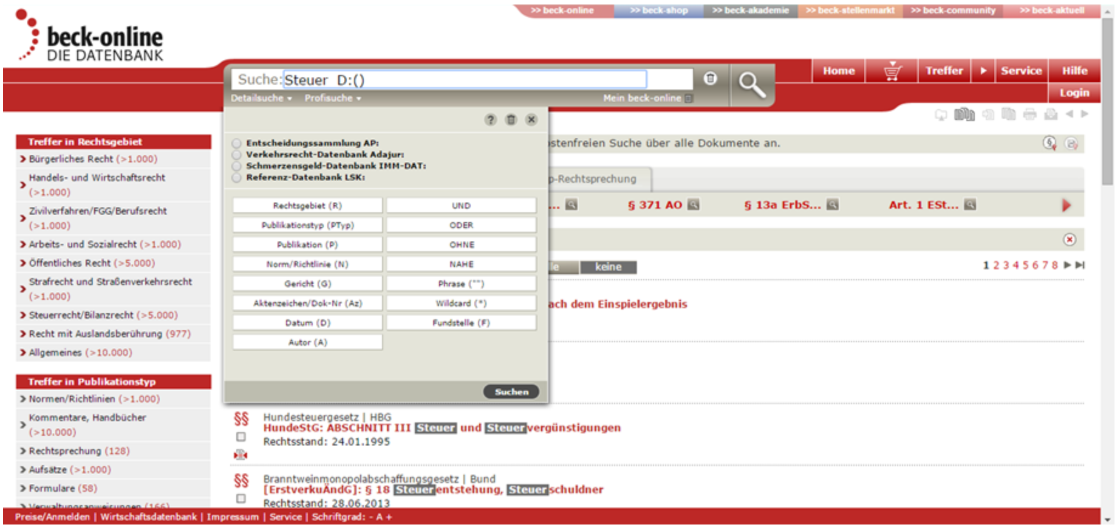


Figure A.8.: Query specification using Boolean or other operators, Wildcard characters and Segment search/query specification using command languages. (Verlag C.H.BECK oHG, 2015)

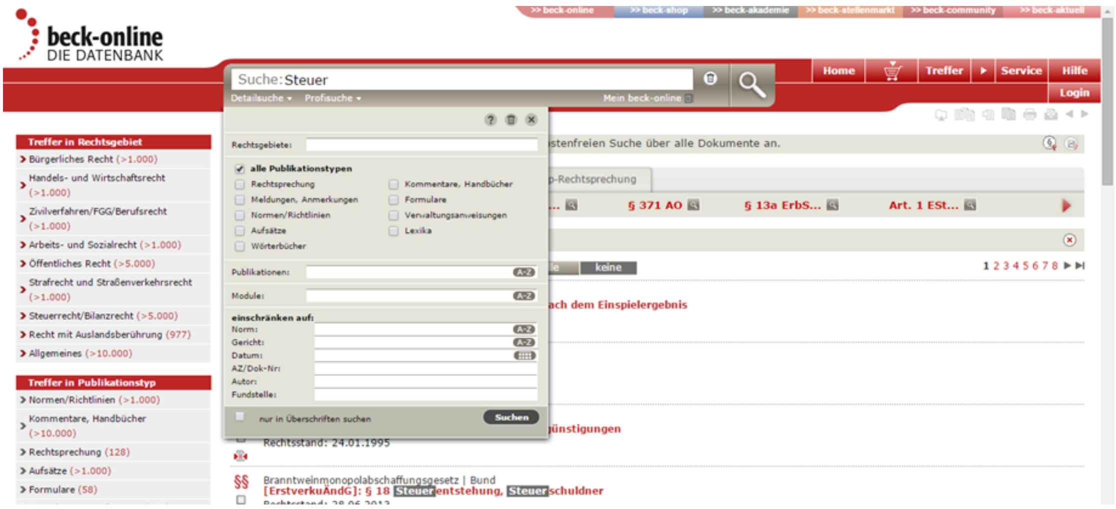


Figure A.9.: Parametric search. (Verlag C.H.BECK oHG, 2015)

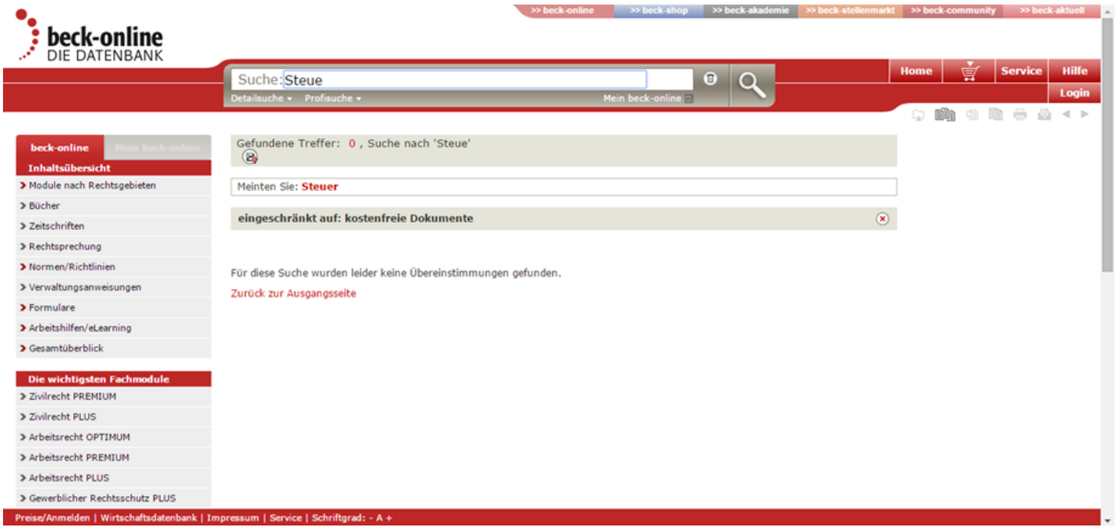


Figure A.10.: Spelling suggestions and corrections and Letter size change (*Schriftgrad*). (Verlag C.H.BECK oHG, 2015)

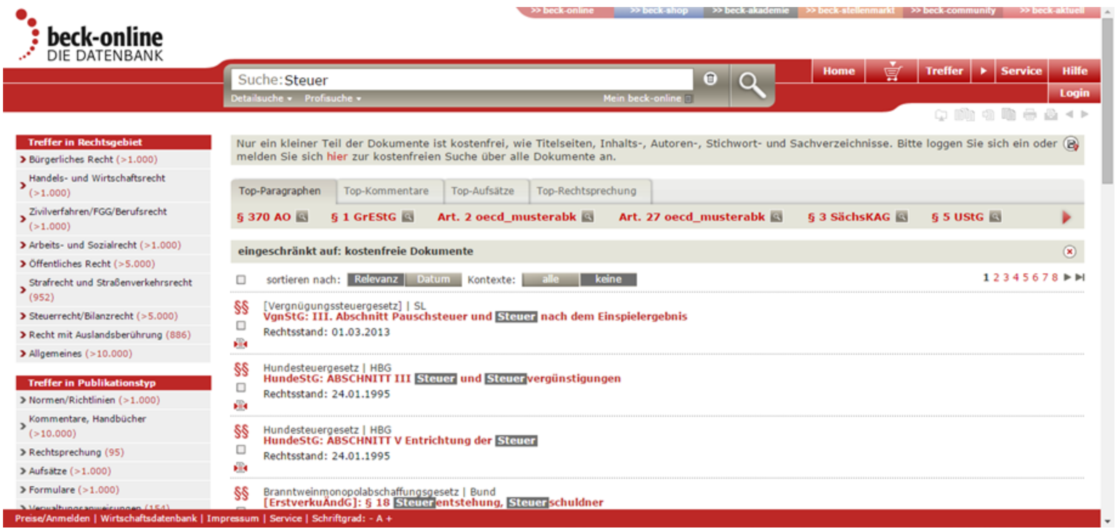


Figure A.11.: Suggesting popular destinations (Top-paragraphs (*Top-Paragraphen*), Faceted navigation (Origin/type of publication (*Treffer in Publikationstyp*), Field (*Treffer in Rechtsgebiet*)), Categories for sorting and filtering search results (Relevance criteria (*sortieren nach Relevanz*), Date (*sortieren nach Datum*)) and Highlight query terms. (Verlag C.H.BECK oHG, 2014a)

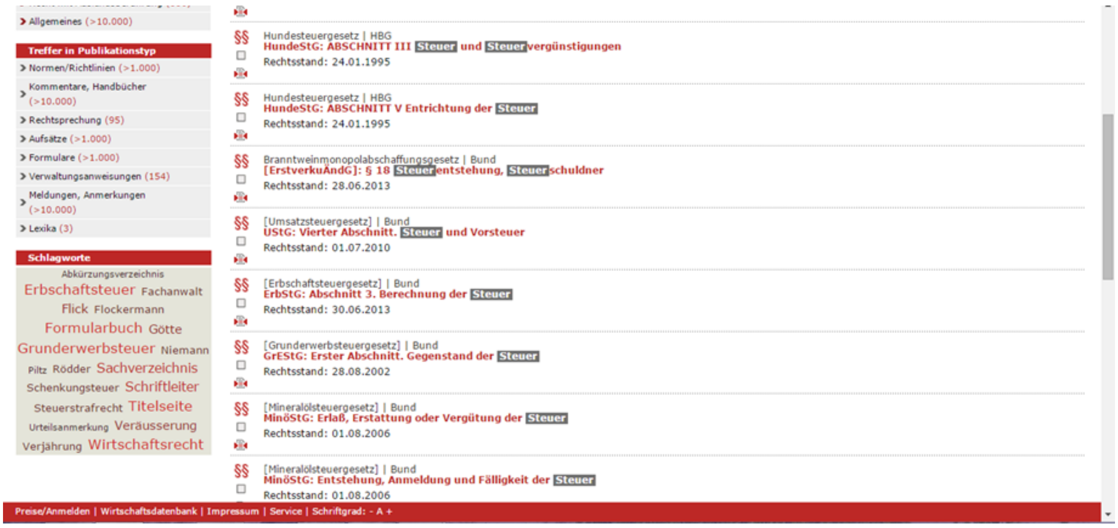


Figure A.12.: Navigating via social tagging and social bookmarking. (Verlag C.H.BECK oHG, 2014a)

A.3. LexisNexis

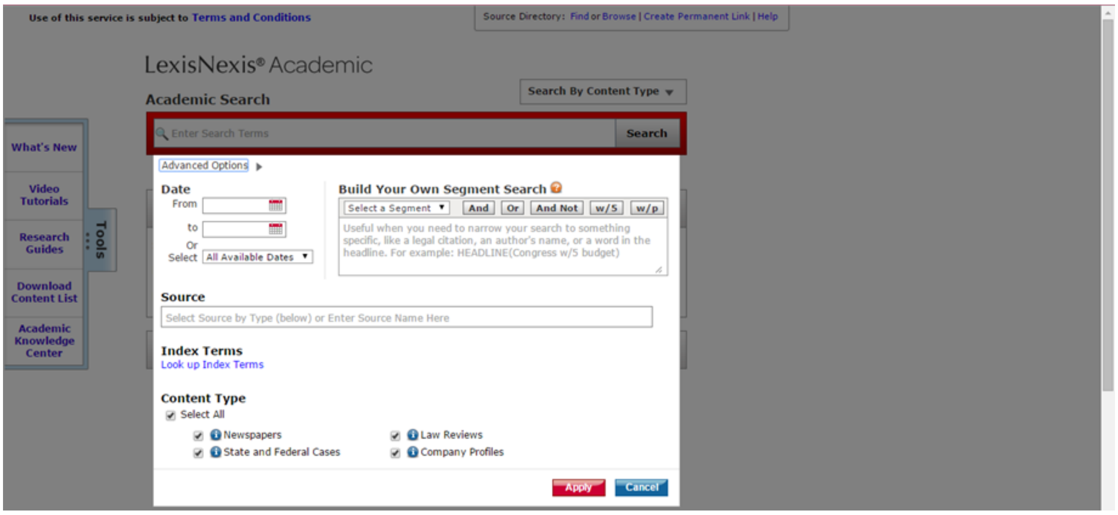


Figure A.13.: Query specification using Boolean or other operators, Segment search/-query specification using command languages and Parametric search/s-coped search. (RELXGroup, 2015c)

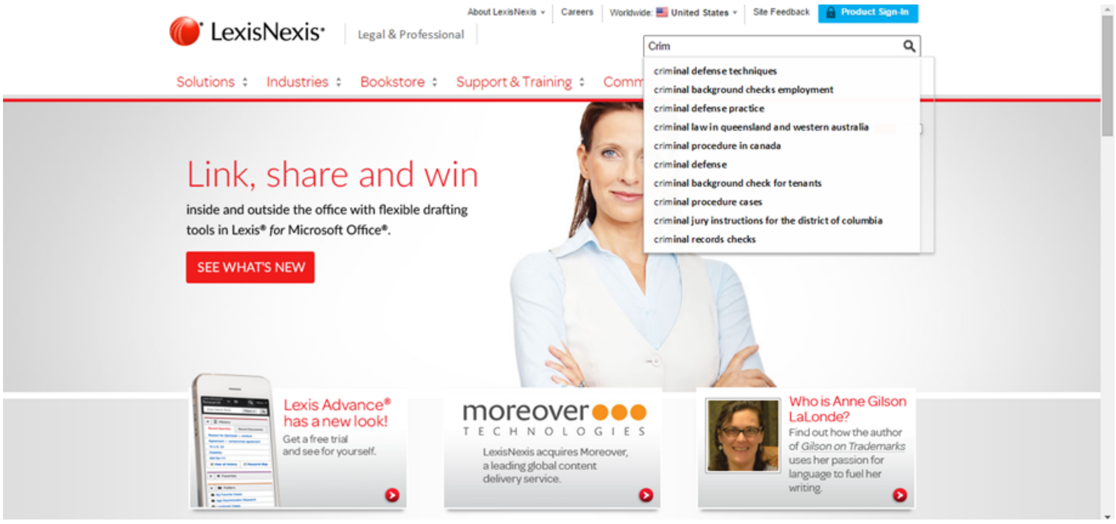


Figure A.14.: Autocomplete. (RELXGroup, 2015b)

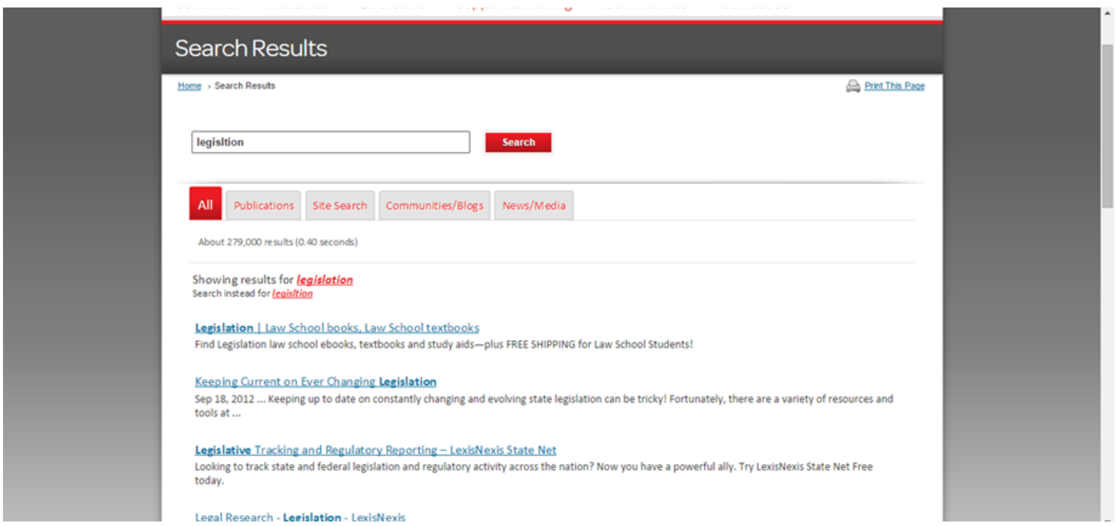


Figure A.15.: Spelling suggestions and corrections, Categories for grouping search results (Origin/type of publication) and Number of matches. (RELXGroup, 2015f)

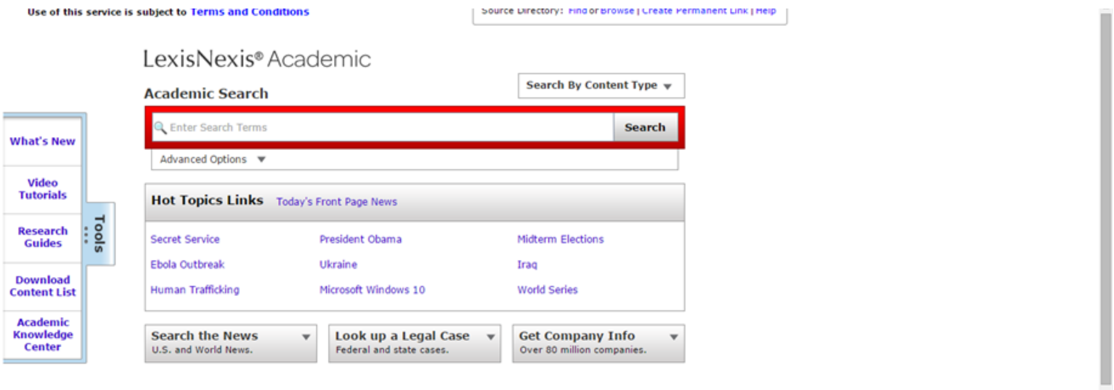


Figure A.16.: Suggesting popular destinations. (RELXGroup, 2015c)

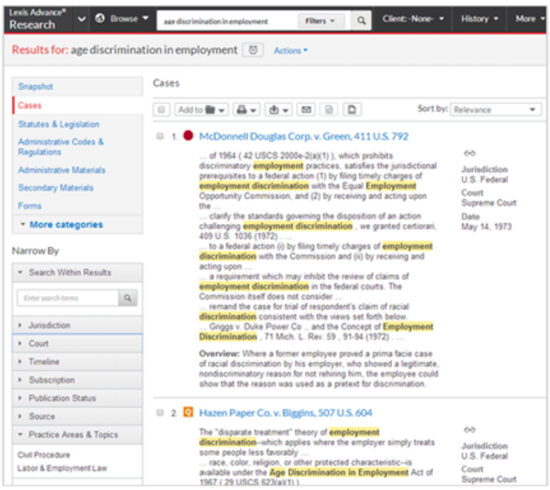


Figure A.17.: Highlight query terms, Categories for sorting and filtering search results (Relevance criteria) and Faceted navigation (Origin/type of publication (Cases Statutes and legislation, etc.), Field (Practice Areas and Topics), Court, Jurisdiction, Timeline, Subscription, Publication status). (RELX-Group, 2015e)

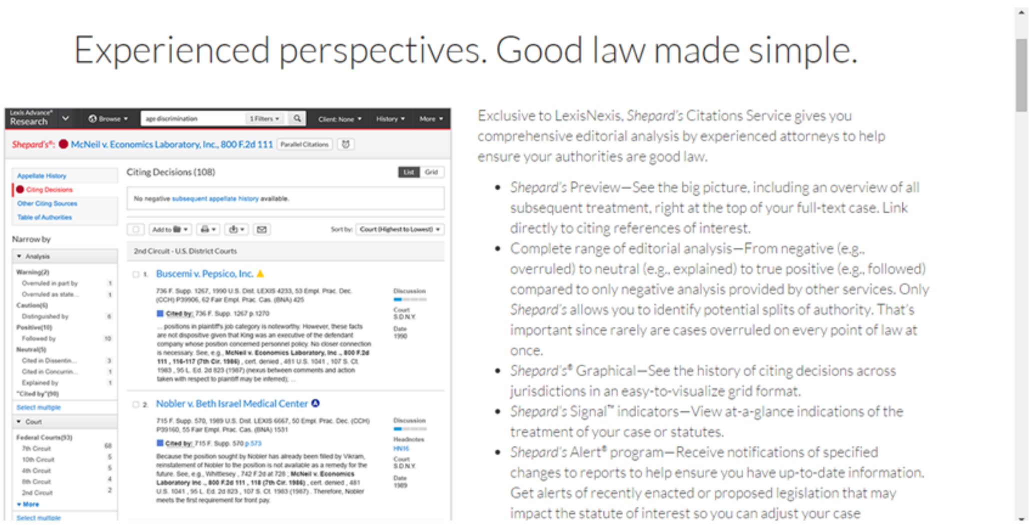


Figure A.18.: Faceted navigation (Citations information). (RELXGroup, 2015d)

A.4. FindLaw

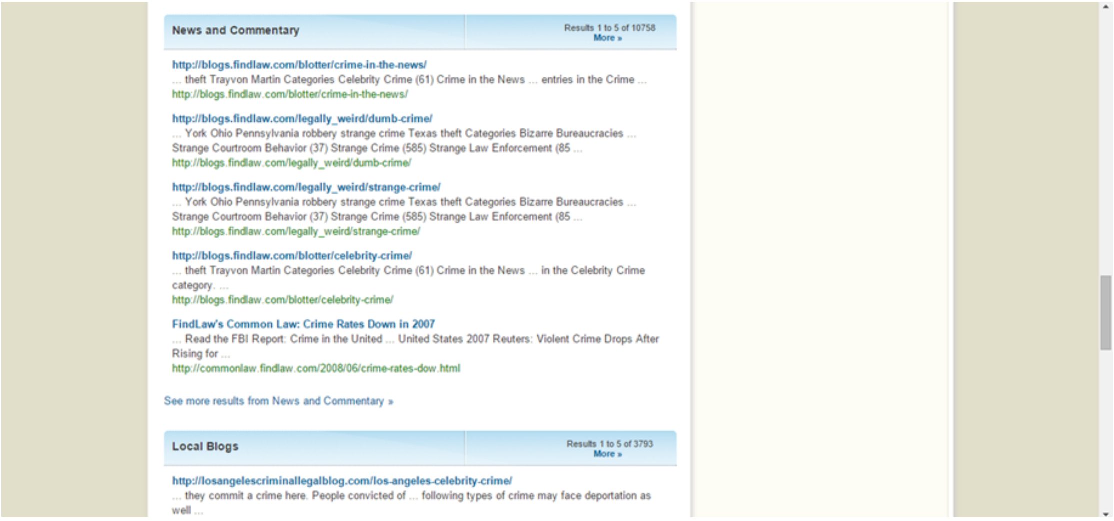


Figure A.19.: Categories for grouping search results (Origin/type of publication) and Number of matches. (Thomson Reuters, 2015b)

A.5. LEXinform

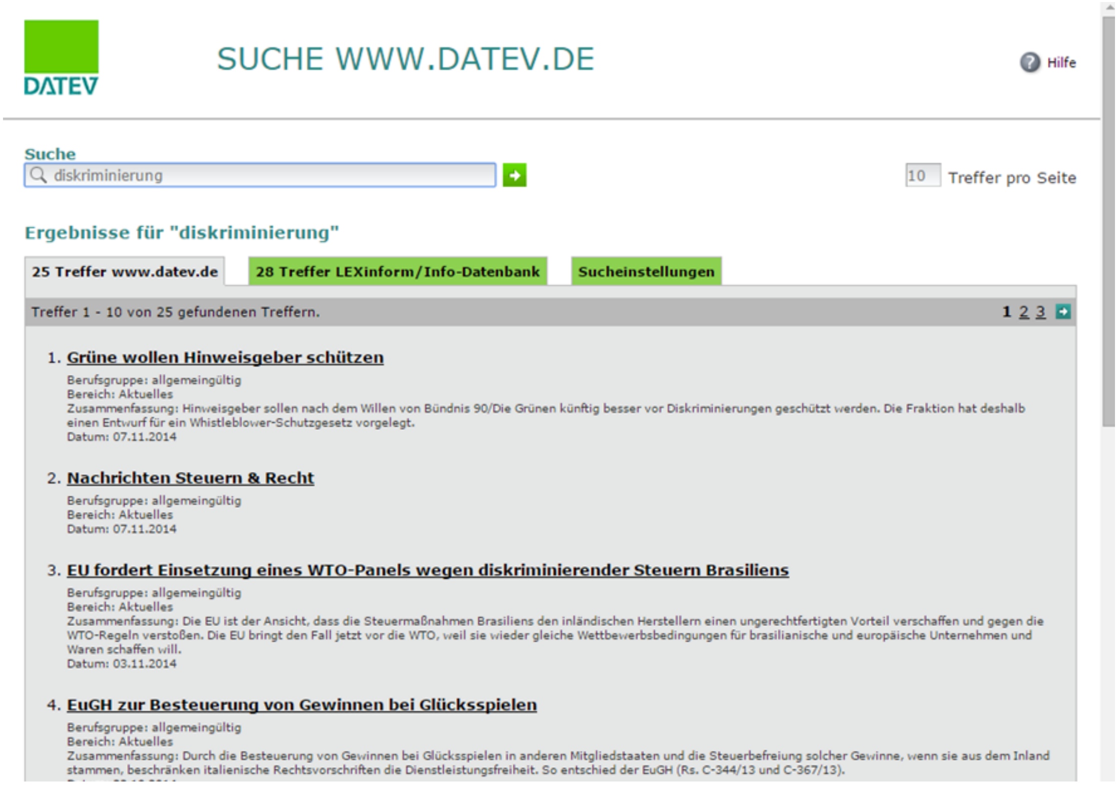


Figure A.20.: Number of matches. (DATEV, 2015b)

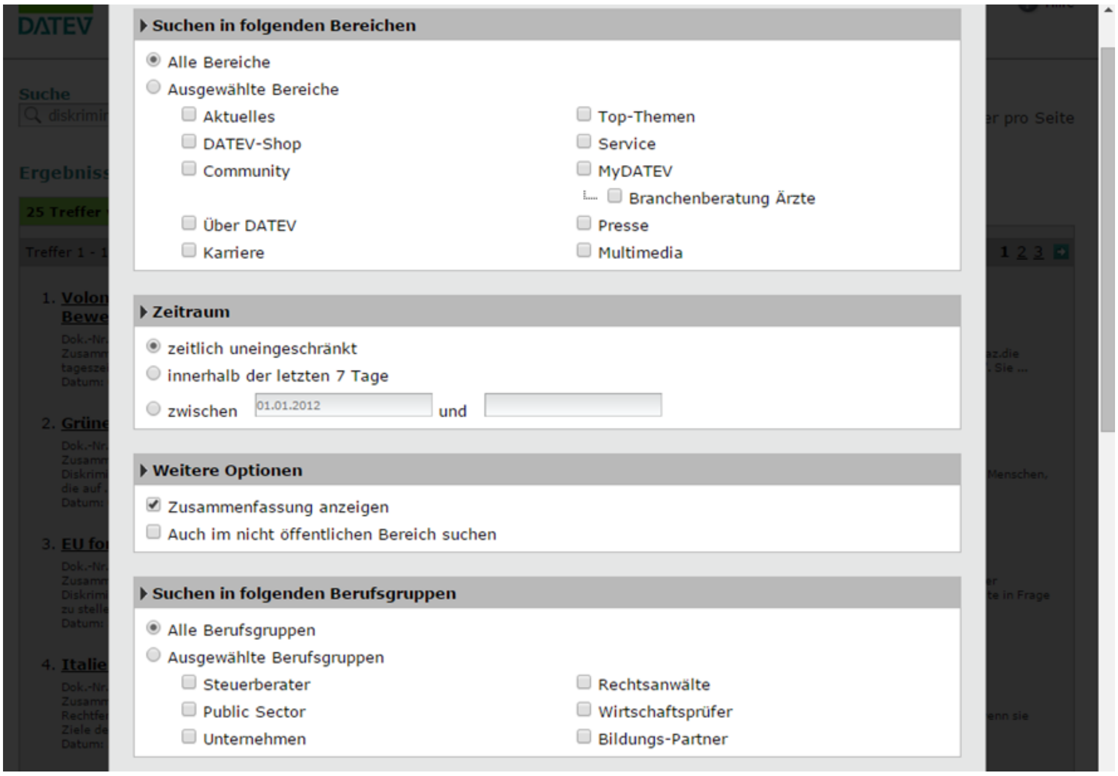


Figure A.21.: Parametric search/scoped search. (DATEV, 2015b)

B. Needs assessment interview

Date: __________ Time: ___________

Who was assessed: ___________________________

Needs assessment interview

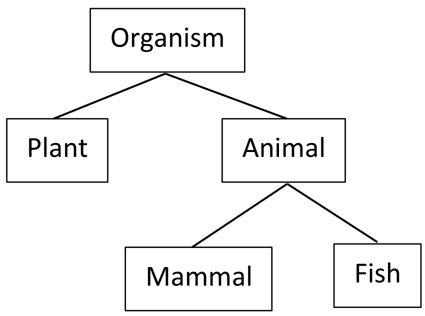
General / Background

1) What is your law area and how much time have you practiced it?

2) Approximately what percentage of your time do you use a computer to perform your job?
    a. 0-25%
    b. 25-50%
    c. 50-75%
    d. 75-100%

3) How often do you review legal literature? (Laws, cases, legal magazines, legal journals, etc…)
    a. Every day
    b. Every week
    c. Every month
    d. Rarely
    e. Never

4) Introduction to the topic
    a. Computer system for legal databases
    b. Lexical knowledge
        i. Example



Date: __________ Time: ___________

Who was assessed: ___________________________

Legal Information Database & Lexical Knowledge

1) Do you use legal information databases? If yes, which one?

2) **In a scale from 1 to 5**, being 5 the highest (100%), do you think the legal information databases will become more important in future?

1	2	3	4	5
○	○	○	○	○

3) Again in a scale from 1 to 5, what score would you give to the next tasks according to how important you think they are during the usage of legal information databases?

	1	2	3	4	5
a. Formulating the query	○	○	○	○	○
b. Exploration	○	○	○	○	○
c. Navigation	○	○	○	○	○
d. Searching	○	○	○	○	○
e. Others? Which one?	○	○	○	○	○

4) Do you think that recommendations, based on your search query and search behaviour, would support your efficiency and effectiveness during searching?

1	2	3	4	5
○	○	○	○	○

5) Could you tell me also in a scale from 1 to 5 how relevant do you think these types of recommendations/tools for searching are?

	1	2	3	4	5
a. Auto-Suggestion	○	○	○	○	○
b. Query refinement	○	○	○	○	○
c. Facetted search (technique for organising content to refine results, e.g. slider for prices)	○	○	○	○	○
d. Selection of Information Source (e.g. cases, laws, comments, etc.)	○	○	○	○	○
e. Others? Which one?	○	○	○	○	○

Date: __________ Time: ___________

Who was assessed: ___________________________

6) Imagine you are searching for the term "Abortion", what kind of information would you like the system to suggest? (What would be useful to you?) (scale from 1 to 5)

	1	2	3	4	5
a. Abortion synonyms (e.g. termination, miscarriage)	○	○	○	○	○
b. Types of abortion (hyponyms) (e.g. spontaneous abortion, induced abortion, vacuum aspiration)	○	○	○	○	○
c. Hierarchical family of the term "Abortion" (parents, hypernyms) i. e.g. Termination→ Change of state→Change→ Action→Human activity →Event→ Psychological feature	○	○	○	○	○
d. Related terms such as malformation, malfunction, discontinuation (coordinate terms, siblings)	○	○	○	○	○
e. Derivate related terms such as abortionist	○	○	○	○	○

7) Suppose you got a case about discrimination against handicap. How useful would you find: (scale from 1 to 5)

	1	2	3	4	5
a. Particular ways of discrimination such as isolate, segregate, disfavour, ghettoize, seclude, put at a disadvantage (troponyms)	○	○	○	○	○
b. Related terms such as distinguish, label, stratify (coordinate terms)	○	○	○	○	○

8) If you got a case about a person having an accident during work and as a result the right hand of the worker was seriously injured. How useful would you find a system that suggests: (also in a 1 to 5 scale)

	1	2	3	4	5
c. Parts of the hand (meronyms) (e.g. digital arteries, intercapitular vein, metacarpal vein, finger, metacarpal artery,…)	○	○	○	○	○
d. The contrary, words that show what the hand is part of, such as… arm, organic structure, human being (holonyms)	○	○	○	○	○

Date: __________ Time: ___________

Who was assessed: ___________________________

Conclusion

9) Do you think such information (like talked before) supports legal searches?

10) Do you think that this information might be too complex to be used by the user?

11) Do you think that the usage of lexical knowledge as provided can meet the semantics (meaning of the language) that legal practitioners have?

12) What kind of support do you really miss in current legal information databases?

13) Do you have any further comments, advices or additional feedback?

List of Figures

List of Tables

Bibliography

Airio, E., Järvelin, K., Saatsi, P., Kekäläinen, J., & Suomela, S. (2004). Ciri-an ontology-based query interface for text retrieval. In *Web intelligence: proceedings of the 11th finnish artificial intelligence conference*.

Angular-UI team. (2014). UI Bootstrap: Bootstrap components written in pure AngularJS by the AngularUI Team. Retrieved March 12, 2015, from http://angular-ui.github.io/bootstrap/

Araujo, Denis Andrei de, Rigo, S. J., Müller, C., & Chishman, R. (2013). Information Extraction from Legal Documents Using Linguistic Knowledge and Ontologies (Unpublished manuscript). Retrieved October 20, 2014, from https://www.academia.edu/6063529/Information_Extraction_from_Legal_Documents_Using_Linguistic_Knowledge_and_Ontologies

Atlassian. (2015). Bitbucket. Retrieved March 18, 2015, from https://bitbucket.org/

Bast, H., Chitea, A., Suchanek, F., & Weber, I. (2007). Ester: efficient search in text, entities, and relations. In C. Clarke, N. Fuhr, & N. Kando (Eds.), *30th international conference on research and development in information retrieval (sigir'07)*. ACM.

Broder, A. (2002). A taxonomy of web search. *ACM SIGIR Forum*, *36*(2), 3. doi:10.1145/792550.792552

Crosby, P. B. (1979). *Quality is free: the art of making quality certain*. New York: McGraw-Hill.

DATEV. (2015a). LEXInform. Retrieved November 10, 2014, from http://www.datev.de/lexinform

DATEV. (2015b). LEXInform: Search Results. Retrieved November 10, 2014, from http://www.datev.de/portal-su/Search.do?source=lexinfodb%5C&query=diskriminierung%5C&pagesize=10%5C&timeframe=any%5C&from=01.01.2012%5C&until=%5C&summary=true%5C&pid=dpi

Deutscher Bundestag. (February 2013). Stichwort: gesetzgebung. Berlin.

Dini, L., Peters, W., Liebwald, D., Schweighofer, E., Mommers, L., & Voermans, W. (2005). Cross-lingual legal information retrieval using a wordnet architecture. In G. Sartor (Ed.), *The 10th international conference on artificial intelligence and law* (p. 163). doi:10.1145/1165485.1165510

Divoli, A., Hearst, M. A., & Wooldridge, M. A. (2008). Evidence for showing gene/protein name suggestions in bioscience literature search interfaces. In *Pacific symposium on biocomputing* (Vol. 13, pp. 568–579).

Elasticsearch BV. (2015a). Elasticsearch. Retrieved March 9, 2015, from http://www.elasticsearch.org/

Elasticsearch BV. (2015b). Elasticsearch: Bulk API. Retrieved March 19, 2015, from http://www.elastic.co/guide/en/elasticsearch/reference/current/docs-bulk.html

Fellbaum, C. (1998.). *Wordnet: an electronic lexical database*. Cambridge, MA: MIT Press.

Fielding, R. T. & Taylor, R. N. (2002). Principled design of the modern web architecture. *ACM Transactions on Internet Technology*, *2*(2), 115–150. doi:10.1145/514183.514185

FullScale Labs, LLC. (2012-2014). Elastic.js. Retrieved March 10, 2015, from http://www.fullscale.co/elasticjs/

Git. (2015). Git: –fast-version-control. Retrieved March 18, 2015, from http://git-scm.com/

Google. (2010-2015a). AngularJS. Retrieved March 12, 2015, from https://angularjs.org/

Google. (2010-2015b). AngularJS: Developer Guide - Services. Retrieved March 25, 2015, from https://docs.angularjs.org/guide/services

Hamp, B. & Feldweg, H. (1997). Germanet - a lexical-semantic net for german. In *In proceedings of acl workshop automatic information extraction and building of lexical semantic resources for nlp applications* (pp. 9–15).

Hearst, M. (2009). *Search user interfaces*. Cambridge: Cambridge University Press.

Henrich, V. & Hinrichs, E. (2010). Gernedit - the germanet editing tool. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, . . . D. Tapias (Eds.), *Proceedings of the seventh international conference on language resources and evaluation (lrec'10)*. Valletta, Malta: European Language Resources Association (ELRA).

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Q*, *28*(1), 75–105. Retrieved March 25, 2015, from http://dl.acm.org/citation.cfm?id=2017212.2017217

Holtzblatt, K. & Beyer, H. (1997). Contextual design. In A. Edwards & S. Pemberton (Eds.), *Chi '97 extended abstracts* (p. 184). doi:10.1145/1120212.1120334

Jothilakshmi, R., Shanthi, N., & Babisaraswathi, R. (2013). A survey on semantic query expansion. *Journal of Theoretical & Applied Information Technology*, *57*(1), 128–138.

Joyent, Inc. (2015). Node.js. Retrieved March 10, 2015, from https://nodejs.org/

JSON. (n.d.). JSON: Introducing JSON. Retrieved March 12, 2015, from http://json.org/

Juris. (2013). Gesetze im Internet. Retrieved February 17, 2015, from http://www.gesetze-im-internet.de/

Matthijssen, L. (1999). *Interfacing between lawyers and computers: an architecture for knowledge-based interfaces to legal databases*. Law and electronic commerce. The Hague: Kluwer Law International.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, *38*(11), 39–41. doi:10.1145/219717.219748

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to wordnet: an on-line lexical database*. *International Journal of Lexicography*, *3*(4), 235–244. doi:10.1093/ijl/3.4.235

O'Hara, T. P. (2005). *Empirical acquisition of conceptual distinctions via dictionary definitions* (Doctoral dissertation, New Mexico State University, New Mexico).

Onyshkevich, B. & Nirenburg, S. (1995). A lexicon for knowledge-based mt. *Machine Translation*, *10*(1-2), 5–57. doi:10.1007/BF00997231

Otto, M. & Thornton, J. (2011-2014). Bootstrap. Retrieved March 12, 2015, from http://getbootstrap.com/

Play. (2014). Play Framework: The High Velocity Web Framework For Java and Scala. Retrieved March 9, 2015, from https://www.playframework.com/

Princeton University. (2010a). Princeton University "About WordNet." WordNet. Retrieved February 25, 2015, from http://wordnet.princeton.edu/

Princeton University. (2010b). Princeton University "About WordNet." WordNet. WordNet 3.0 Reference Manual - Glossary of WordNet terms. Retrieved February 16, 2015, from https://wordnet.princeton.edu/man/wngloss.7WN.html

Pustejovsky, J. & Bergler, S. (1992). *Lexical semantics and knowledge representation: first siglex workshop, berkeley, ca, usa ... 1991 : proceedings*.

RELXGroup. (2015a). LexisNexis. Retrieved November 7, 2014, from http://www.lexisnexis.com/

RELXGroup. (2015b). LexisNexis. Retrieved November 7, 2014, from http://www.lexisnexis.com/en-us/home.page

RELXGroup. (2015c). LexisNexis: Academic. Retrieved November 10, 2014, from http://www.lexisnexis.com/hottopics/lnacademic/

RELXGroup. (2015d). LexisNexis: Advance. Good Law. Retrieved November 10, 2014, from http://www.lexisnexis.com/en-us/products/lexis-advance/feature-good-law.page

RELXGroup. (2015e). LexisNexis: Advance. Right Results. Retrieved November 10, 2014, from http://www.lexisnexis.com/en-us/products/lexis-advance/feature-right-results.page

RELXGroup. (2015f). LexisNexis: Search Results. Retrieved March 9, 2015, from http://www.lexisnexis.com/en-us/search.page

Saravanan, M., Ravindran, B., & Raman, S. (2009). Improving legal information retrieval using an ontological framework. *Artificial Intelligence and Law*, *17*(2), 101–124.

Schweighofer, E. & Geist, A. (2007). Legal query expansion using ontologies and relevance feedback. In *Loait* (pp. 149–160).

Smith, J. C. (1997). The use of lexicons in information retrieval in legal databases. In *Proceedings of the 6th international conference on artificial intelligence and law* (pp. 29–38).

The Apache Software Foundation. (2011-2012). Apache Lucene. Retrieved March 12, 2015, from http://lucene.apache.org/

Thomson Reuters. (2015a). FindLaw. Retrieved November 10, 2014, from https://www.findlaw.com/

Thomson Reuters. (2015b). FindLaw: Search Results. Retrieved November 10, 2014, from http://public.findlaw.com/LCsearch.html?entry=crime%5C&restrict=consumer%5C&button=SEARCH+AGAIN

Typesafe, Inc. (2015). Typesafe: Activator. Retrieved March 18, 2015, from http://typesafe.com/activator

University of Tübingen. (2014). GermaNet - A German Wordnet: GermaNet - An Introduction. Retrieved February 27, 2015, from http://www.sfs.uni-tuebingen.de/GermaNet/

Verlag C.H.BECK oHG. (2014a). Beck-online: Die Datenbank. Retrieved November 7, 2014, from https://beck-online.beck.de

Verlag C.H.BECK oHG. (2014b). Beck-online: Die Datenbank. Retrieved April 10, 2015, from https://beck-online.beck.de

Verlag C.H.BECK oHG. (2015). Beck-online: Die Datenbank. Retrieved March 9, 2015, from https://beck-online.beck.de

Waltl, B. & Matthes, F. (2015). Supporting the legal subsumption process: determination of concreteness and abstractness in german laws using lexical knowledge.

Wolters Kluwer. (2014a). Jurion: Einfach besser zu Recht finden. Retrieved November 7, 2014, from https://www.jurion.de/

Wolters Kluwer. (2014b). JURION: Einfach besser zu Recht finden. Retrieved March 9, 2015, from https://www.jurion.de/

YousefED. (2014). ElasticUI: AngularJS directives for Elasticsearch. Retrieved March 10, 2015, from http://www.elasticui.com/