Chapter 3

Information retrieval

3.1 General information retrieval

The science of information retrieval deals with the problem of accurately and speedily retrieving relevant information from vast information storages.[163] Ever since computing advanced to the point beyond mere calculating machines, to the state where computers had capacity to store non-trivial amounts of information in structured or full text form, they have been used to do information retrieval tasks.

The problem of how to efficiently find information is older than computers, however. In libraries, systems using index cards have been used to catalogue all items in the library. An index card would contain a number of properties for a specific item, such as author, title, publisher, year, classification in some classification scheme and information on how the item could be retrieved (i.e. on what shelf it was placed).[164] A complete index system for a library would have two or more sets of cards, sorted on different properties, i.e. one sorted by author's surname and another sorted by the classification scheme. These systems support the most rudimentary forms of information retrieval. Particularly, it makes it possible to retrieve an item that is already known to exist, if some identifying property (or combination of properties) of it is known. If you know the author and title, you can look up the author in the set of cards that are sorted by author, and then look through each card until the correct title is found, and on that card read on which shelf the item is stored.

At first, computer systems did not have storage enough to store the full text of documents. Furthermore, the process of digitizing existing documents required that the text be keyed in by hand. The earliest IR systems

[163] C. J. van Rijsbergen: Information Retrieval, 2nd ed., London 1979.

[164] The most well-known such scheme is probably the Dewey Decimal System, which uses a decimal number for each possible classification of an item. By adding decimals, new more specific classifications can be created underneath existing broader categories.

therefore mimicked index card systems by storing only basic properties about each item, along with information of where the full text could be retrieved. The primary advantage of these basic systems was that a single set of cards (or *records*, the term used for storing information about a single item in a computer system) could support finding information stored in any property that was present on the records, and that any number of users could use the index simultaneously, even from remote locations.

The process of storing information about a document in such a form that it can later support finding out about the document and retrieving it is known as indexing. Information retrieval research quickly advanced past emulating the index card systems of physical libraries, and taking advantage of the computers' ability to store and search information.

Still, the general mindset of early digital information retrieval was rooted in library science. One indication if this is that one of the earliest formulations of the problem of storing and retrieving information is known as the *library problem*, as coined in a 1960 paper by Maron and Kuhns.[165] It describes a technique for indexing and searching literature in a "mechanized library". Their solution involves storing a set of tags for each documents, where each tag was a key content-bearing word, and then formulating a mathematical relationship between the set of tags in a document with the set of tags in an information request called the *relevance number*, intended to represent the probability that the document was relevant with respect to the information need.

In Maron and Kuhns model, only a few tags for each document were stored. The indexing process had to be done by hand. Soon, computer storage and processing capabilities made it possible to store the entire text of documents, which opened up the possibility of retrieving the entire text, not just a location where the hard copy could be found. It also opened up the possibility of indexing the document automatically. Instead of choosing a few indexing terms or tags for each document, it became possible to compile a list of each individual word contained in any of the stored documents, save for a number of very common connectors such as "the", "an", "of" and similar, known as *stop words*. This method of retrieval is known as *full text retrieval*. The list of words (alternatively known as the dictionary, vocabulary or lexicon of the entire corpus) is then stored together with information of in which documents that word can be found. The list of locations of a single word is known as a *postings list* (or inverted list) as it lists each individual occurrence (posting) of the word, and the compilation of the entire dictionary and all the posting lists are referred to as an inverted index.[166]

[165] M. E. Maron/J. L. Kuhns: On Relevance, Probabilistic Indexing and Information Retrieval, in: Journal of the ACM 7.3 (July 1960), pp. 216–244.

[166] Manning/Raghavan/Schüze: Introduction to information retrieval (see n. 101), p. 6.

With these building blocks, we can create a simple information retrieval system.[167] We express the information need in one or more terms, and then use the inverted index to find those documents that contain these terms. This is the basic foundation of information retrieval. Although the process of indexing and searching has been developed tremendously, the building blocks of indexing still look similar. In particular, the notion of a dictionary and a postings list for each term in that dictionary is still used. This notion can support a number of different retrieval models (although the exact implementation of these indices may differ depending on model).[168]

3.1.1 Retrieval models

Regardless of whether we use a system where terms for documents have been carefully selected by hand, or if we index the documents automatically to create thousands of terms for each document, we need some way of expressing a need for information and defining what documents should be retrieved in order to satisfy that need. We need a *retrieval model*. A retrieval model can be defined by how it represents documents to be retrieved, how it represents the queries used to retrieve documents, and by which method a query is matched by zero or more documents.[169]

A retrieval model specifies only the high level structure of how queries, documents and matching functions are represented. For each retrieval model, decades of research has gone into refining different aspects of query and document representation, as well as different formulations of matching functions and relevance feedback functions. Ideas formulated within one model (e.g. different methods for term weighting or approaches for user-aided query reformulations) acts as inspiration for further research within other models. Models can also be combined, such as creating an initial result set using boolean search, which is then ranked using the vector space model.[170]

Common to all retrieval models are the concept of a result set. This is

[167] In order to keep this description brief and focused on the relevance aspect of retrieval models, we have ignored computational linguistic techniques such as stemming or lemmatization of terms, that is, reducing words to their stem or base form, which is required to make a query for "searching" match a document containing the word "searched" by reducing both of these to the base form "search". We have also omitted techniques for recognizing and indexing multi-word terms such as "information retrieval". For a comprehensive introduction to these concepts, see Karlgren: Information Retrieval: Statistics and Linguistics (see n. 102), sec. 3.2

[168] Manning/Raghavan/Schüze: Introduction to information retrieval (see n. 101), sec. 4.6.

[169] Marie-Francine Moens: XML Retrieval Models for Legislation, in: Legal Knowledge and Information Systems. Jurix 2004: The Seventeenth Annual Conference, 2004, p. 2.

[170] An early system using this approach was the Syracuse Information Retrieval Experiment (SIRE), described in Gerard Salton/Edward A. Fox/Harry Wu: Extended Boolean Information Retrieval, in: Communications of the ACM 11.26 (Nov. 1983), pp. 1022–1036, here p. 1023

a subset of the entire corpus, consisting of those documents that are found to match the query. In document retrieval, the objects in this result set is always documents themselves, not parts of documents, information from particular documents, or synthesized data from several documents. This can be contrasted with fact retrieval, in which the result set (or rather "answer") consists of facts extracted from one or more documents in the entire corpus. Unless otherwise indicated, the below text only refers to document retrieval.

The set can be more or less delimited. In boolean retrieval, every document in the corpus is either clearly in the result set for a particular query or clearly not. In other models such as the vector space model or probabilistic models, documents are part of the result set to a certain degree.

This fundamental property of the retrieval model, or rather its matching function, can be described as being either "exact match" or "best match".[171] Another way of describing the same thing is that the matching function can be either an "identity function" or a "nearness function".[172]

For best-match-functions, some form of threshold value can be used to specify a limit for the degree of matching. The criteria used in these models for determining whether something is part of the result set can also be used as the basis for ranking the documents in some order. But the criteria for ranking can be based on other things, such as publication date, document type or weighted zone scoring (assigning different weight to different terms of the document, depending on where in the document they are present).

In the following, three large classes of retrieval models will be covered (boolean, vector space and probabilistic, respectively). This is not the only way of classifying retrieval models (one could add rule-based, cluster-based, connectionist and semantical or logical models[173]), but it is a commonly used classification.

Boolean retrieval models

Boolean retrieval is the simplest and the most predictable of the retrieval models. Queries in this retrieval models consist of terms and connectors. The simplest possible query has only a single term. The result of such a query is the set of all documents that contain the term. If the user wants to search for two or more terms, it must first be decided whether the wanted result set is the set of documents containing all of the terms (using the AND connector) or the set of documents that contains any of the terms (using the OR connector).[174]

[171] Rijsbergen: Information Retrieval (see n. 163), p. 1.
[172] Bing: Handbook of Legal Information Retrieval (see n. 12), pp. 161.
[173] Turtle: Text Retrieval in the Legal World (see n. 17), p. 23.
[174] Manning/Raghavan/Schüze: Introduction to information retrieval (see n. 101), sec. 1.1.

More advanced queries can be constructed by grouping terms and connectors using parentheses. As an example, to find cases on internet defamation, the user wants documents that contain "internet" or any equivalent term, as well as "defamation" or any equivalent term. The final query can then be "(internet OR web OR online) AND (defamation OR slander OR harassment)". The query is processed by handling the parenthesized expressions first (creating two disjunct sets of documents, each being the union of three sets, one for each basic term), then creating the final result set (by constructing the intersection of those two sets).[175]

The usage of terms in the description above ("disjunct sets", "union", "intersection") reflects the boolean retrieval model foundations of set theory. The result set is unordered, i.e. all documents match the query in an equal amount. There is, in the basic boolean model, no way of ranking the documents according to relevance or similarity to the query. The notion of relevance in the basic boolean model is that of a binary property where a document is either relevant to a query or it is not.

If the information is available, it is possible to sort the result set according to date of publication, author, document type, etc, which in many cases can be good enough.[176]

There are several ways of ranking result sets based on the query in boolean search systems. We will briefly describe two approaches for this.

Weighted zone scoring:[177] Many types of documents consists of several distinct parts. A scientific paper will usually have a title, an abstract, the main body of text, and a concluding list of references. A legal case will have a headnote and the main text of the decision (both of which can be broken up in yet smaller parts). By considering each such part an independent zone, and applying the boolean query to each in part, we get a set of match / no-match results for each zone. By assigning different weights to different zones, such as the weights add up to 1.0, we can calculate a final score between 0 and 1 for each document with respect to the boolean query. If our corpus documents all have a title (weight 0.4), an abstract (weight 0.3), a body (weight 0.2) and a references zone (weight 0.1), and that a particular document only satisfies the query for the title and body zones, the score for that document will be 1*0.4 + 0*0.3 + 1*0.2 + 0*0.1 = 0.6.

This mechanism can be further extended to allow each zone to score

[175] A more evolved implementation of this strategy, where the user would enter a series of synonym groups, each group forming a conceptor, was researched in the NORIS project at NRCCL in the early 1980s. This strategy allowed for basic ranking of the result set, see Jon Bing: Text Retrival in Norway, in: Program 15.3 (1981), pp. 150–162, here pp. 157

[176] One interesting example is the AustLII search interface (at http://www.austlii.edu.au/), which allows for sorting the result set according to citation frequency, so that documents that are often cited by other documents occurs before documents that are less often cited

[177] Manning/Raghavan/Schüze: Introduction to information retrieval (see n. 101), sec. 6.1.

values other than 0 or 1 by allowing for partially successful boolean matches (such as counting a zone which have three out of four required terms as 0.75 instead of 0).

It can also be used for documents without clearly delimited zones, for example assigning the weight 0.5 for the first 10% of the document, and then lower weights for successive 10% segments, on the theory that terms that are central to the topic of the document will appear early in the document.

Extended Boolean retrieval:[178] The terms used in a query will often have different relative importance for expressing the information need. Likewise, the terms contained in a document will have different importance for distinguishing documents. Very common terms (such as "law" or "right" in a legal document corpus) will be present many times in almost all documents. More specific terms (such as "reunification" or "inadmissibility") will be present in relatively few. It makes sense to consider terms that are common in a particular document but uncommon in the corpus as a whole – i.e. terms that are "uncommonly common"[179] – to have more weight when ranking documents. In the extended boolean model, these weights are used to calculate a *similarity* between the query and the document.

In order to calculate this weight, we need to introduce a pair of metrics that have proven to be useful not just for extended boolean retrieval, but information retrieval and text processing in general: the *term frequency* (TF) and the *inverse document frequency* (IDF).[180] There are a number of different ways of calculating these, but in the simplest case, $TF_{t,d}$ is simply the count of how many times the term t occurs in document d. IDF is calculated by first calculating the document frequency (DF_t) by counting how many documents the term occurs in out of the collections total N documents, and then calculating $IDF_t = log\frac{N}{DF_t}$.

IDF_t thus becomes a measure of how unlikely it is that a document would feature a certain term. Once we have these values, we can calculate the TF-IDF value as $TF\text{-}IDF_{t,d} = TF_{t,d} \times IDF_t$. The TF-IDF value for term t and document d is highest when it occurs many times in that document, but few times in the corpus as a whole – i.e. when it is "uncommonly common".

The extended boolean retrieval model can be seen as a hybrid between the exactness of boolean retrieval and the sophisticated term weighting of the vector space model, described below. In fact, the algorithm has a constant p whose value can vary from 1 to ∞ , the effect of which is to create results more like the standard boolean model (when $p = \infty$), or more like vector space models (when $p = 1$).[181]

Variations of extended boolean searching have been central in legal in-

[178] Salton/Fox/Wu: Extended Boolean Information Retrieval (see n. 170).

[179] Karlgren: Information Retrieval: Statistics and Linguistics (see n. 102), sec. 3.1.2.

[180] For a complete description of TF and IDF, including different normalization schemes, see Manning/Raghavan/Schüze: Introduction to information retrieval (see n. 101), sec. 6.2

[181] Salton/Fox/Wu: Extended Boolean Information Retrieval (see n. 170), p. 1025.

formation retrieval, for example being the default search method (called "Terms and connectors") in the Westlaw system.[182]

Vector space models

As stated earlier, a retrieval model is defined in part by how it represents documents and queries. In the standard boolean model, the documents are simply represented by the inverted index (the collection of the dictionary and the postings lists of the system). In the vector space model (VSM), this index can still be used, but the main document representation is that of a vector in a multidimensional document space.[183]

The concept of multidimensional vectors can perhaps best be understood by considering three-dimensional vectors and extrapolating from this. A vector is a direction in a space. We commonly think of regular space as three dimensional, but mathematically, a space can have any number of dimensions.

Consider an IR system whose corpus contains three (very short) documents:

- A: "retrieval"
- B: "legal information"
- C: "legal information retrieval"

As the vocabulary of this system has three terms, it can be described by three-dimensional vectors:

Document	"legal"	"information"	"retrieval"
A	0	0	1
B	1	1	0
C	1	1	1

By comparing these vectors, we can see which documents are more like each other. In this simple example, it is easy to see that document B is more like document C (two of three terms in common) than it is like document A (no terms in common). For larger document collections, both the construction of vectors, and the comparison of them, gets more complicated.

Regarding the construction of vectors, just having each dimension being either 0 or 1 is not expressive enough. A document where a term occurs 20 times should have a higher value for that term (dimension) than a document

[182] Manning/Raghavan/Schüze: Introduction to information retrieval (see n. 101), p. 15.
[183] G. Salton/A. Wong/C. S. Yang: A Vector Space Model for Automatic Indexing, in: Communications of the ACM 18.11 (Nov. 1975), pp. 613–620.

where the term occurs only once. In practice, variants of TF-IDF weighting are used to calculate a value for each term in a document vector.[184]

Regarding comparison, the standard way of comparing two vectors is to calculate the *cosine distance* (or cosine similarity) between the two vectors x and y having M terms:[185]

$$\frac{\sum_{i=1}^{M} x_i \times y_i}{\sqrt{\sum_{i=1}^{M} x_i^2} \times \sqrt{\sum_{i=1}^{M} y_i^2}}$$

The numerator is the result of the *dot product* of the vectors, while the denominator is the product of the *euclidean length* of each vector.

Other similarity measures than cosine distance can be used. In 1982, Tapper experimented with a similarity measure specific for case law documents, where the elements of the vectors to be compared are based on cited and citing case (*citation vectors*), instead of indexed terms.[186]

As we can see, the use of vector space models are not restricted to information retrieval, but can be used for e.g. *document clustering*, i.e. automatically classifying documents (either in predetermined categories or by constructing categories from how similar documents cluster together).

In fact, as presented so far, there has been no notion of how to represent queries, only documents. We will describe query representations below, but first, we should observe an important thing about document similarity: "closely associated documents tend to be relevant to the same requests", also known as the *cluster hypothesis*.[187] Methods for clustering or categorizing documents can therefore immediately have applications in information retrieval in that they allow us to expand a initial result set, presumably one with high precision and lower recall, into a larger result set that improves recall without sacrificing (much) precision.[188]

But without a starting document, how do we go from our information need to a result set? By expressing our information need in the form of a (short) document, which is then converted to a query vector (using similar, but not necessarily identical term weighting as the document vectors), and then comparing this to all available document vectors.[189]

[184] Manning/Raghavan/Schüze: Introduction to information retrieval (see n. 101), sec. 6.2.2.

[185] Ibid., sec. 6.3.1.

[186] Tapper: An experiment in the use of citation vectors in the area of legal data (see n. 158).

[187] Rijsbergen: Information Retrieval (see n. 163), p. 30.

[188] A similar approach based on boolean models is described in Benny Brodda: Gimmie More O' That. A Potential Function in Document Retrieval Systems?, in: Peter Seipel (ed.): From Data Protection to Knowledge Machines. The Study of Law and Informatics, 1990, pp. 251–270

[189] Manning/Raghavan/Schüze: Introduction to information retrieval (see n. 101), sec. 6.3.2.

As noted, the assumption in VSM based models is that similar documents have a tendency to be relevant to each other (when regarding the query in the same way as a document). This in turn assumes a notion of relevance as a continuous function between queries and documents. A document can be more or less similar to a query, and thus more or less relevant to the query.

Probabilistic models

The boolean model can be extended by ranking the result set. Probabilistic models, in contrast, has no fixed result set. Instead the entire document corpus is ranked according to the probability ranking principle.[190]

This principle was first formulated by M.E. Maron and J. L. Kuhns in 1960 in the following terms:[191]

If $P(A.I_j, D_1) > P(A.I_j, D_2)$, then D_1 is more relevant than D_2.

It was later reformulated in a way that might be easier to understand by van Rijsbergen:[192]

> "If a reference retrieval system's response to each request is a ranking of the documents in the collection in order of decreasing probability of relevance to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data have been made available to the system for this purpose, the overall effectiveness of the system to its user will be the best that is obtainable on the basis of those data."

As a principle, it seems sound. But how do we estimate the probability that a particular document is relevant to a request? There are a number of different ways to do that.

We will cover one of the more basic probabilistic models, known as the *Binary Independence Model*. This model (BIM for short) attempts to estimate the probability of a document being relevant to a query using Bayes theorem and a number of simplifying assumptions.

Both document and query are viewed as simple vectors (like in VSM) where each dimension in the vector is either 1 (meaning the term is present one or more times in the document or query) or 0 (meaning the term isn't present). This simple representation accounts for the "Binary" in BIM. The other simplifying assumption, which it shares with VSM, is that document

[190] Ibid., sec. 11.2.1.

[191] Maron/Kuhns: On Relevance, Probabilistic Indexing and Information Retrieval (see n. 165), p. 221.

[192] Rijsbergen: Information Retrieval (see n. 163), p. 88.

terms are considered to be independent of each other. This is an assumption shared with many other models, known as the "bag-of-words" assumption, and accounts for the "Independence" in BIM.

The process of ranking a document set using BIM consists of calculating a *retrieval status value* (RSV) for each document and query. Calculating this requires two estimates:

- $\frac{p_t}{(1-p_t)}$: The odds of a term t appearing in a document that is relevant to a query. This is often set to 0.5 initially, but may change as we learn more about the ratio of relevant documents to nonrelevant (e.g. by relevance feedback mechanisms).

- $\frac{u_t}{(1-u_t)}$: The odds of a term t appearing in a document that is nonrelevant to a query. This is often set to DF_t/N, that is the document frequency for t (i.e. the number of documents that the term appears in) divided by the total number of documents.[193]

Once we have these odds, we can calculate the individual *odds ratio* (the odds that the term appears in a relevant document, divided by the odds that it appears in an irrelevant document) for each term, and then calculate RSV for document d and query q, where $w_{t,d}$ is the document vector dimension for term t (1 if present, 0 otherwise) and $w_{t,q}$ is the same for the query:[194]

$$RSV_d = \sum_{t=1}^{M} w_{t,d} \times w_{t,q} \times log(\frac{p_t(1-u_t)}{u_t(1-p_t)})$$

It should be noted that when using the suggested value for $\frac{p_t}{(1-p_t)}$, the weighting function $log(\frac{p_t(1-u_t)}{u_t(1-p_t)})$ (also known as the RSJ-weight, from its authors Robertson and Spärck-Jones) becomes a variant of inverse document frequency (IDF).[195]

As stated above, BIM makes a number of assumptions; the assumption that terms appear independently of each other is hardly correct.[196] Nevertheless, it works well in practice, although a number of other probabilistic models perform better.[197] Nowadays the BM25 algorithm (commonly named Okapi BM25) is considered state-of-the-art in probabilistic

[193] Manning/Raghavan/Schüze: Introduction to information retrieval (see n. 101), sec. 11.3.3.

[194] Equation from ibid., sec. 11.3.1, adjusted for easier comprehension after the example in Martin Eriksen: Rocchio, Ide, Okapi och BIM. En komparativ studie av fyra metoder för relevance feedback, MA thesis, Högskolan i Borås, 2008, p. 17

[195] Ibid., p. 17.

[196] Manning/Raghavan/Schüze: Introduction to information retrieval (see n. 101), sec. 11.3.

[197] Fabio Crestani et al.: "Is This Document Relevant?...Probably": A Survey of Probabilistic Models in Information Retrieval, in: ACM Computing Surveys 30.4 (Dec. 1998), pp. 528–552 contains a survey of commonly used models

retrieval.[198]

Based on this introduction to probabilistic models, we can say a few things about what assumptions it makes about relevance.

The user expresses an information need through a set of terms. From these, a relevance number (e.g. RSV) can be calculated. This relevance number has a continuous scale. Unlike boolean retrieval, probabilistic retrieval has ranking built-in. But like boolean retrieval, it assumes a binary notion of relevance. The relevance number represents the probability of the document being relevant to the information need, not the extent of its relevance to same.

3.1.2 Link analysis

As we have seen, the three basic families of retrieval models have, at its core, rather simple idea of what relevance is. For boolean retrieval, a document is relevant to the query if its terms place it in the set of documents that the query specifies. For vector space models, similarity between a document and a query acts as a proxy for relevance between the same. For probabilistic models, the relevance probability is based on relative term frequencies and term weights in the document, the corpus and the query.

In all three models, we try to represent the query and the documents as an unordered set of terms.[199] These terms can be an unstructured concordance, automatically extracted, of the words contained in the documents or they can be a carefully engineered hierarchical taxonomy of important concept in whatever domain the documents are concerned with.[200] A document can be indexed using both these approaches (automatically, by extracting all terms from the full text of the document, and manually, by having a domain expert read and classify it). Yet neither of these approaches is infallible. It is hard to find synonyms or adapt to changes in language when automatically extracting terms.[201] Manual (intellectual) indexing is inherently subjective, and even the same indexer may sort the same document under different terms depending on which context the document is presented in.[202]

There are other aspects of documents besides which terms they contain (or can be indexed under). Examples of such aspects are publication dates

[198] Manning/Raghavan/Schüze: Introduction to information retrieval (see n. 101), sec. 11.4.3.

[199] In some advanced probabilistic models, term ordering may be considered

[200] An example of such an engineered taxonomy is the Eurovoc classification scheme, which contains over 6000 terms or *descriptors*. Much EU legal information such as case law and secondary law is manually indexed using Eurovoc.

[201] Although not impossible – according to the *distributional hypothesis* words with similar meanings tend to occur in the same context. The technique of *random indexing* can be used to find synonyms in this way. See e.g. Magnus Sahlgren: An introduction to random indexing, tech. rep., SICS, Swedish Institute of Computer Science, 2005

[202] Bing/Harvold: Legal Decisions and Information Systems (see n. 7), p. 41.

and authors. Such properties can be used to narrow searches with very high precision, if the user's information need can be expressed using these properties (if a user wishes to see legal cases from after a certain point in time, it is trivially easy to construct a query system that only returns such cases). This is an example of how to use non-term properties of documents to create better IR systems. Another approach is to make use of how documents refer to each other.

The nature of references

A reference is any sort of mention of one document (or a certain part of a document) in the text or metadata of a document. The referencing and the referred document may be one and the same, in which case we may talk about internal references. A document may make any number of such references, and the referenced documents may in turn refer to yet other documents. Together, the set of documents form a graph or a network which we call a *citation network*. References are always directed, i.e. it has a source and a target, and these two are not interchangeable. This means that the resulting graph is a directed graph. In most settings, references are time-bound in that they can only be made from a newer document to an older document, and not in the other direction (as the newer document did not exist when the older document was written). The exception for this is documents that can be updated or revised, as the revision can incorporate references to newer documents (but of course, such an updated document can be considered newer than the document it points to, bringing us back to strict time-bound references). In these cases, the result is a *directed acyclic graph*, i.e. a graph with no cycles in that it is impossible to start at any document, follow outgoing references and end up at the starting document.

In order to be useful as a reference, it must be unambiguous, i.e. it must be possible to uniquely identify the document that is referenced. Under this definition, mentions such as "Saracevics seminal article" or "The supreme court's landmark civil rights decision" are not references in themselves, although they may be put in context in a way that make the targets uniquely identifiable. When references are used in information retrieval, the requirement for restriction is made more stringent as the automated IR system must be able to interpret the reference in order to use it.

References may be so implicit that they require significant interpretation and background knowledge from the reader. When it comes to legal information, this opens up questions of the authority of references, particularly when a certain interpretation is fixed in the form of a machine-readable reference in the digital manifestation of a source, whereas the printed source had an implicit reference.[203]

[203] Sjöberg: Critical Factors in Legal Document Management (see n. 45), pp. 146.

How references are used

Some examples of the use of references:

- **In academic publishing**, authors make references to earlier research (in the form of papers, dissertations and similar documents) to indicate the origin of information, methods and ideas used.[204]

- **In case law**, courts make references to earlier cases dealing with substantially similar issues. These are used to justify the decision, both when deciding in the same way or, when some key facts differ, deciding in another (*distinguishing* the case).

- **On the web**, authors refer to other resources (pages) by creating links to the resources' location (URL). These are used for a multitude of reasons, but they all enable visitors to retrieve the referenced resource by a simple click.

What references indicate

A reference is in most cases a form of endorsement. When an academic refer to an earlier paper, this most often indicates that the content of that paper is scientifically sound and contains useful information. When a court cites an earlier case they primarily do it in order to justify their decision on the basis of the earlier case. And when a web page author creates a link to another page, thereby making it easy for readers to access the page, the action implies a recommendation.

But this is not always the case. Academics may refer to earlier research to criticize its methodology or refute the claims made. Courts may cite a case in order to overturn the precedent set therein. And web authors may disagree with the content of the page they're linking to.

The assumption that a reference implies endorsement is at the foundation of bibliometrics. As discussed in sec. 2.3.2, bibliometry deals with using citations in academic publications to create rankings of journal impact factor. A number of limitations of or problems with that approach have been identified, including obliteration through incorporation, self-citations, negative citations, and the effect of document age and literature size.[205]

Such problems are common in other citation networks as well. But as the success of Google and PageRank show, they are not problems of such magnitude that they make citation analysis unworthwhile.

[204] David Easly/Jon Kleinberg: Networks, Crowds, and Markets: Reasoning about a Highly Connected World, 2010, p. 378.

[205] Geist: Using Citation Analysis Techniques For Computer-Assisted Legal Research In Continental Jurisdictions (see n. 162), pp. 72.

Aspects of the citation network

As mentioned, the references in our three examples form a graph, where each document is a node, and each reference is an edge. Graph theory has long studied networks of many kinds, including information networks such as the ones described above.

First, we must observe, with respect to our earlier discussions about different types of relevance (sec.2.3.1, ranging from objective to subjective relevance), that references are an objective manifestation of a subjective judgment. Each author decides, with at least some measure of subjectivity, which sources to reference. But once cited, there exist an objective record of this subjective decision. The citation network in any document collection then becomes an objectively measurable property of a series of subjective assessments.

Secondly, in general network theory, there are two classes of networks that are of particular interest. A *small-world network* is defined as a network where the distance for two random nodes, measured in number of hops along edges of connected nodes, grows proportionally to the logarithm of the number of nodes in the network. This means that even in very large graphs, the distance between two nodes is surprisingly small.[206]

The degree distribution of a network is a measure of how the edges in the network are distributed between the nodes. If the distribution is one where a few nodes have a large number of connections, and a large number of nodes have few or no connections, the distribution follows a power law. Such a network is known as a *scale-free network*. By contrast, a random network would have a bell-curve degree distribution.[207]

These classes are interesting because of the mechanisms that build them. Why is a particular web page often linked (it may be because it is well written and about a topic interesting to many, or it may be because it ranks highly in search engines for a common keyword search, and thus is discovered by many)? Why is a particular case often cited (it may be because it answers an open question on how to classify certain facts of the case, or because it formulates a new legal rule).

By looking at properties of the resulting graphs, we can corroborate hypotheses about the network-building mechanisms.

[206] The name "small-world network" is derived from the concept "small world phenomenon", the observation that in social networks, the distance between people is surprisingly small. For more on the small world phenomenon, see Easly/Kleinberg: Networks, Crowds, and Markets: Reasoning about a Highly Connected World (see n. 204), p. 611

[207] Geist: Using Citation Analysis Techniques For Computer-Assisted Legal Research In Continental Jurisdictions (see n. 162), p. 61.

Analyzing the citation network

Regardless of the properties of the citation network as a whole, analyzing the citations themselves can be useful when ranking results. Using citations as a base for IR had been studied in the 80s,[208] but it was the advent of the web that really initiated research into citation (or rather link) analysis on a large scale.

Behind the idea of link analysis is the assumption that each link is a form of endorsement. As we have discussed above, this assumption do not always hold, but in general it's a reasonable starting point. A simple algorithm based on this would be the InDegree-algorithm, which simply ranks documents according to the number of other documents that contains citations to them.[209] For many reasons, this approach does not scale to the entire content of the web, so what is needed is some way of distinguishing between endorsements that carry different weight.[210]

In the late 1990s', three such suggestions were made almost simultaneously:

- Hyperlink Vector Voting (HVV):[211] This model is based on the vector space model, but instead of using terms extracted from documents, it uses terms extracted from links to documents. The more links that exists to a document (and more diversity in link anchor texts), the greater the probability of that document ranking high in a query.

- Hyperlink-induced Topic Search (HITS):[212] This model calculates two scores for each document - the hub score (representing its value as a useful resource list) and the authority score (representing its value as an authority on its topic). These two scores are seldom high for the same page, but a page with high authority need to be referenced by many pages with high hub score, and vice versa.

- PageRank:[213] This model calculates a single value for each document, calculated in an iterative fashion. Conceptually, there is a finite amount of this value (known as Pagerank) for the entire graph, and all pages

[208] See e.g. C. Tapper: The use of citation vectors for legal information retrieval, in: Journal of Law & Information Science 1.2 (1981), pp. 131–161, W. Bruce Croft/Howard Turtle: A retrieval model incorporating hypertext links, in: Proceedings of the second annual ACM conference on Hypertext (HYPERTEXT '89), New York, NY, USA 1989, pp. 213–224

[209] Allan Borodin et al.: Link Analysis Ranking: Algorithms, Theory, and Experiments, in: ACM Transactions on Internet Technology 5.1 (2005), pp. 231–297, here p. 234.

[210] Manning/Raghavan/Schüze: Introduction to information retrieval (see n. 101), sec. 21.

[211] Yanhong Li: Toward a Qualitative Search Engine, in: IEEE Internet Computing 2 (4 1998), pp. 24–29.

[212] Jon M. Kleinberg: Authoritative Sources in a Hyperlinked Environment, in: Journal of the ACM 46 (1999).

[213] Page et al.: The PageRank Citation Ranking: Bringing Order to the Web. (See n. 159).

initially have the same amount. For each iteration, the value for each page gets redistributed to all pages that it links to (divided by the number of links). Pagerank values thus "flow" within the link graph until an equilibrium is reached.[214] This model is the most well known by association with the Google search engine, which used this model initially. The exact algorithm used by Google today is not generally known, but it is widely believed to be a HITS-derived algorithm known as Hilltop.[215]

These models (particularly HITS) have been subject to much improvement and research.[216] HITS have also been applied to case law citation networks from the US Supreme Court with good results.[217]

3.1.3 Evaluation of information retrieval

IR systems are developed and used in order to help users find relevant information. They have been subject to much research and improvement for half a century. But how can we tell whether they are getting better? Indeed, what does "better" even mean for an IR system?[218]

The first, and still dominant, approach for evaluating the performance of IR systems was the so-called Cranfield experiments in the late 1950s.[219] The resulting methodology is known as the Cranfield paradigm, and is built upon a series of predefined document collections, information needs (known as Topics) and a set relevance judgments for these documents and topics known as the *gold standard* (or, alternatively, ground truth). Using these, it is possible to measure a number of key metrics from the system in an automated and precise fashion. The most important of these are precision and recall. A simple description of these is as follows:[220]

Precision refers to how many of the presented documents are actually relevant in the particular context (e.g. to a search query, or in a "related documents" browsing interface element). If the system presents ten documents, and three of these are relevant, the system has a precision of .3.

[214] There are a number of other steps, primarily to avoid the problem of "sinks" (nodes that are linked to, but does not in turn link to other pages) capturing all Pagerank available in the system

[215] Easly/Kleinberg: Networks, Crowds, and Markets: Reasoning about a Highly Connected World (see n. 204), p. 413.

[216] See e.g. Longzhuang Li/Yi Schang/Wei Zhang: Improvment of HITS-based Algorithms on Web Documents, in: WWW2002, 2002, pp. 527–535 and Borodin et al.: Link Analysis Ranking: Algorithms, Theory, and Experiments (see n. 209)

[217] James H. Fowler/Sangick Jeon: The authority of Supreme Court precedent, in: Social Networks 30 (2008), pp. 16–30.

[218] C.f. p. 10

[219] Manning/Raghavan/Schüze: Introduction to information retrieval (see n. 101), sec. 8.8.

[220] Rijsbergen: Information Retrieval (see n. 163), p. 114.

Recall refers to how many relevant documents are presented in a particular context (e.g. to a search query, or in a "related documents" browsing interface element) - if there are ten relevant documents in the system, and eight of these are displayed, the system have a precision of .8.

It has often been observed that precision and recall are opposite metrics. A system that has high precision will have low recall, and vice versa. In order to have a single metric that emphasizes the trade-off between precision and recall, the F-measure was developed by van Rijsbergen.[221] It is the weighted harmonic mean of precision and recall.[222]

The above measures work fine for evaluating boolean, non-ranked systems. But for ranked systems, it becomes more crucial that the results near the top really are relevant. There have been a number of measures proposed for evaluating ranked sets. The most common metric today is the mean average precision (MAP).[223] It is a metric for an entire series of information needs (topics), which consists of the mean of the average precision for each single information need. The average precision in turn is calculated by taking the precision of the top k results for a single information need, and then calculating the average of these as k varies from 1 until whatever value is needed to retrieve all relevant results. Graphically, the average precision can be thought of as the size of the area under the precision/recall curve.

If the relevance judgments in the gold standard are graded, instead of binary, the above metrics cannot take that extra information into account. *Rpref* is an evaluation metric, based on the earlier *bpref*, which take graded judgments into account.[224]

The Cranfield paradigm has been criticized from many aspects. It assumes a very system-oriented view of relevance,[225] and the fact that it uses expert-curated information needs and corresponding relevance judgments instead of real users masks the fact that what we really want to know about a system is its ability to satisfy actual users. Real users seldom know how to construct an optimal query for a system, and their satisfaction depend on a number of things that are more difficult than recall, MAP and f-measures to measure. Particularly user interface considerations can have a large effect on satisfaction. It is possible to evaluate user satisfaction, but it us much more difficult (and expensive).[226] The Cranfield paradigm thus remains the

[221] In it's original form, it was known as the effectiveness measure, defined in ibid., p. 134 as the complement of what today is known as the F-measure, see ibid., sec. 8.8

[222] Manning/Raghavan/Schüze: Introduction to information retrieval (see n. 101), sec. 8.3.

[223] Ibid., sec. 8.4.

[224] Jan DeBeer/Marie-Francine Moens: Rpref - A Generalization of Bpref towards Graded Relevance Judgments, in: SIGIR '06, 2006.

[225] Birger Hjørland: The Foundation of the Concept of Relevance, in: Journal of the American Society for Information Science and Technology 61.2 (Feb. 2010), pp. 217–237, here p. 218.

[226] Manning/Raghavan/Schüze: Introduction to information retrieval (see n. 101),

predominate evaluation methodology in information retrieval.[227]

3.2 Legal information retrieval

3.2.1 History

It is not surprising that legal information retrieval systems have been developed and studied for a long time. Interest in efficient legal information retrieval predates actual automated legal information systems, as evidenced by e.g. the publication Shepard's Citations, which started in 1873[228] and still lives on. The publication, which essentially is a database of citations between cases in US courts, sorts these citations by cited case. A lawyer who, in court filings, wishes to cite a legal rule stated in an older case, needs to make sure that the principle formulated in the older case has not been superseded or distinguished by later cases. It is also important to know if it has been extended by later cases. By following the citations listed in Shepard's Citations, the 1873 lawyer could make sure that ones' legal arguments were up to date. This practice is so common that it has become a verb - to "shepardize a case" means to follow these citations to discover later cases that may further develop the legal rule(s) presented in the first case.

In 1945, Bush wrote the influential essay "As we may think",[229] which describes a futuristic vision of the information tools that the author expected to develop in the coming years, and what they would mean for the scientist, the knowledge worker and society in general. The main tool described was called the Memex, which would use microfilm for information storage and a navigation system which enabled the machine to navigate through thousands of volumes. In particular, the concept of "trails", or the process of connecting information contained in different volumes, is reminiscent of today's hypertext based information systems.

Inspired by Bush's vision, the lawyer Kelso speculated what such a device would mean for the practice of law. In an essay the following year, he described a specialization of the Memex machine, in Kelso's vision dubbed Lawdex.[230] In particular, he envisioned that "all decisions, all new text books, rules, regulations, statutes, commercial data, accounts of relevant political facts, and the like" would be published in a standardized microfilm-based format that could be used directly by the machine.[231] He also envisioned that publishers of this data would, when preparing it, evaluate the

sec. 8.6.2.

[227] Chris Buckley/Ellen M. Voorhees: Retrieval Evaluation with Incomplete Information, in: SIGIR '04, 2004.

[228] Garfield: Citation Indexes for Science - A New Dimension in Documentation through Association of Ideas (see n. 103), p. 108.

[229] Vannevar Bush: As We May Think, in: The Atlantic, July 1945.

[230] Kelso: Does The Law Need a Technological Revolution? (See n. 29).

[231] Ibid., p. 388.

weight of it, "so that a lawyer can, if he prefers, review only *four-star data* on the problem – i.e., data of the greatest weight in controlling the problem."[232]

While some of the aspects in these early visions, such as the usage of analogue microfilm, seem antiquated, other aspects, such as the interactive creations of user trails between different texts, or the associative organization of knowledge, still feels like the cutting edge of information management.

It would not take long until the first applications of computer-based information systems to the problem of laws. In 1956, Horty and Kehl ran a project to study and improve the health statutes of Pennsylvania. A newly passed bill mandated that the phrase "retarded child" be changed to "exceptional child" in all statutes. In order to solve this problem, the project entered all statute text on punch cards, and then performed what essentially was a keyword search to find all the places in the texts where the word "retarded" preceded the word "child".[233]

The following decades of research and development of legal information systems have been described in great detail by e.g. Bing and Harvold in 1977,[234] and again by Bing in 1984.[235] We refer to these texts for further information about the early history of legal IR.[236]

3.2.2 The legal reasoning process

The process of retrieving legal information (henceforth the legal IR process) takes place within a context where the IR activity is supportive of some larger process. These larger processes have been described in slightly differing ways by Bing[237] and Wahlgren.[238] The steps in this process can be uniformly characterized as the following:

1. **Identification of legally relevant facts**: After having established that there exist a problem which may have a legal nature, a lawyer (or any other legal expert) extracts the legal problem of the situation. A legal problem can be described as any problem for which a legal argumentation can contribute to a solution. After isolating the legal aspects of the problem, the lawyer establishes probable and proven facts of the case. Note that even though relevancy with respect to facts

[232]Ibid., p. 390.
[233]Bing: Handbook of Legal Information Retrieval (see n. 12), p. 257.
[234]Bing/Harvold: Legal Decisions and Information Systems (see n. 7).
[235]Bing: Handbook of Legal Information Retrieval (see n. 12).
[236]They are freely available online at *http://www.lovdata.no/*
[237]Primarily in Bing: Handbook of Legal Information Retrieval (see n. 12), but see also Bing/Harvold: Legal Decisions and Information Systems (see n. 7) and Jon Bing: Legal Decisions and Computerized Systems, in: Peter Seipel (ed.): From Data Protection to Knowledge Machines, 1990, pp. 223–250
[238]Primarily in Wahlgren: Automation of Legal Reasoning - A Study on Artifical Intelligaence and Law (see n. 15) but see also idem: Legal Reasoning: A Jurisprudential Model, in: idem (ed.): Scandinavian Studies in Law vol. 40. Legal Theory, 2000

of the case is a crucial step for high quality legal reasoning, this is a different form of "relevance" as compared to its meaning in information retrieval, legal or otherwise. This step is termed *Identification* by Wahlgren,[239] and *Introduction* and *The facts of the case* by Bing.[240]

2. **Search for relevant legal norms**: The identified facts are used as input for the retrieval process. The goal of the retrieval process is to find legal norms that can advance or support a legal argument for arriving at a certain decision. Legal norms are not the same thing as legal sources, but the latter can be used when arguing that the former exists. In any legal system there are certain meta-norms (the doctrine of legal sources) which govern what sources can be used for this. These meta-norms designate a collection of legal statements (statutes, prejudicial cases, preparatory works and the like) which are texts from which norms can be construed. It is this collection of texts that information retrieval technologies can search. This step is termed *Law search* by Wahlgren,[241] and *Legal sources* and *The retrieval process* by Bing.[242]

3. **Interpretation of rules**: After having found textual sources that can form the basis of an argument, the next step is to interpret the exact meaning (or possible meanings) of these rules. The problems and possibility of legal interpretations are far too numerous to describe here, as well as out of scope for the topic of this thesis. But we must be aware that interpretation, like search, is bound by legal meta-norms, and that these may have effect on how search is performed. In particular, the reconciliation (harmonization) of seemingly conflicting norms is often based on the differing importance of different sources. This step is termed *Interpretation* by both Wahlgren[243] and Bing.[244]

4. **Applying the rules to facts**: After having a set of interpreted rules, it is in theory a simple step to apply the rules to the established facts at hand in order to arrive at a decision. This step is termed *Rule application* by Wahlgren,[245] and *The normative interval* by Bing,[246]

[239] Wahlgren: Automation of Legal Reasoning - A Study on Artifical Intelligaence and Law (see n. 15), p. 153.
[240] Bing: Handbook of Legal Information Retrieval (see n. 12), pp. 7.
[241] Wahlgren: Automation of Legal Reasoning - A Study on Artifical Intelligaence and Law (see n. 15), p. 172.
[242] Bing: Handbook of Legal Information Retrieval (see n. 12), pp. 13.
[243] Wahlgren: Automation of Legal Reasoning - A Study on Artifical Intelligaence and Law (see n. 15), p. 188.
[244] Bing: Handbook of Legal Information Retrieval (see n. 12), p. 26.
[245] Wahlgren: Automation of Legal Reasoning - A Study on Artifical Intelligaence and Law (see n. 15), p. 203.
[246] Bing: Handbook of Legal Information Retrieval (see n. 12), p. 37.

5. **Evaluating the result**: The decision process, like the legal IR process in itself, is iterative and may need to be repeated based on what has been learned. By evaluating the result of the rule application, the lawyer can feed back this knowledge into a new repetition of the entire process. This step is termed *Evaluation* by Wahlgren,[247] and *The result - and feedback from the result* by Bing.[248]

The subject of this thesis is, in a nutshell, about ways to make step 2 (*The search for relevant legal rules*) more effective so that step 3 (*Interpretation of rules*), can be faster, simpler and with less uncertainty.

3.2.3 The legal information retrieval process

When we are new to a subject, we don't often know enough to formulate useful queries. Harvold recently attributed the success of Google in general information retrieval to the fact that the queries rarely contain much that we can use to infer the users actual needs from. This information must be found elsewhere.[249]

Even though the user may have the same basic problem through the whole research process (e.g. determining the validity of a proposition of law - a problem which is independent of any particular person tasked to solve it), the users information need shifts through the process. To start with, the user needs background knowledge of the subject area. In many cases, parts of the retrieval processes will be performed intuitively as the lawyers' memory and training enables the lawyer to find the relevant area of law, statutes and precedent. Once comfortable with the basic concepts, the user can attempt to do an initial matching of facts of the case with the most prominent concepts of the law and norms regarding those concepts.

The main retrieval process begins with the user formulating a query that attempts to find norms which have *conditions* that matches the facts of the case (and, if arguing for any particular solution, with *consequences* that results in that solution). The query will, in a sense, be a representation of the facts of the case.

The result of executing this query will be a set of legal sources. From these, the lawyer will interpret or extract the legal norms. If applicable legal norms cannot be found in the sources (or rather, if not *all* applicable norms can be found), the lawyer will reformulate the query. It's this iterative process that is the core of legal research.

As the user finds one or more norms that can potentially be used, the information need shifts to become more concerned with finding examples or

[247] Wahlgren: Automation of Legal Reasoning - A Study on Artifical Intelligaence and Law (see n. 15), p. 217.
[248] Bing: Handbook of Legal Information Retrieval (see n. 12), p. 40.
[249] Harvold: Is searching the best way to retrieve legal documents? (See n. 38).

definitions, some sort of assurance that the user's thinking so far is consistent with established legal practice in the field. During this process, one or more coherent legal argument strategies may form in the users mind by interpreting, applying and evaluating the found norms, possibly also identifying new relevant facts and rules. At the end, the information need has shifted yet again and the user now tries to find possible problems with the tentative strategy by looking for exceptions to the rule. If the user tries diligently and fails, the strategy can reasonably be relied on. The paradox of legal information retrieval is that it's not until the end of this process that the user understands the subject area well enough to formulate the queries that was needed right at the outset.[250]

Since the information need shifts throughout this entire process, the relevance of each potential document shifts as well. This may seem to be an argument that there only exists subjective relevance. But actually, these information needs are quite often not unique (to the user, the problem and the legal sources at hand), but rather generic such as "I need an overview of concepts in labor law" or "What are the possible allowed ways of terminating an employment". Even when getting down to the last stage of the retrieval process, chances are that information needs like "Is harassment against co-workers always a valid ground for dismissal?" have been well studied and that there exists a list of resources that may objectively be considered relevant for them.

At the time when Bing described the model of the legal decision process, information drought was the problem. [251] Nowadays, the problem is the opposite – physical factors determining availability are all but eradicated. It is actually possible to have the total volume of legal sources relevant to the case available in electronic versions. The problem instead becomes one of information overload.

The fundamental aspect of Bing's model is that there is a difference between legal sources and legal norms. Legal sources can be expressed in text and retrieved using computer based or manual methods. But it's the legal norms that can be used in the legal argument, which at the end of the day is the core of legal work.

3.2.4 Using citation networks in legal information retrieval

Link analysis has proved to be useful in general IR. The citation between cases form a citation network, which has many similarities with hyperlink networks that link analysis algorithms have been applied to. How has this citation network been used in legal IR?

[250] Peter Seipel: Juridik och IT. Introduktion till rättsinformatiken, 8th ed., 1997, p. 175.

[251] *"Obviously, the lawyer does not have access to the total number of legal sources, but only to the part of these sources which is available to him."* Bing: Handbook of Legal Information Retrieval (see n. 12), p. 16

We have already described a prominent system started in the analouge IR era, that is Shepard's Citations. As IR systems went from card index to reference retrieval to full text retrieval, it was suggested that the structure of legal information could be used in these systems. In 1970, Marx suggested that the citation network formed by court cases could be used to select relevant cases.[252] The method suggested was to create automatic lists of citing cases and presenting these in conjunction with the case cited – a feature which now is common in legal IR systems.[253] The ideas were also tested by Marx on a subset of US supreme court cases.

Another early influential suggestion (that has already been mentioned) was Tappers idea that citations to and from cases could be used to calculate a similarity measure between cases.[254] Like retrieval using the vector space model, each case was represented as a vector, but based upon citing and cited cases instead of terms appearing in the document (citation vectors). Particularly interesting from a legal point of view was that Tapper did not use the standard cosine distance function to calculate similarity, but rather a custom function designed to take into account aspects of citation practices that are particularly distinguishing – for example, citations to very old cases, cases in other jurisdictions (some of the example cases were from US federal courts, which may cite case laws from other states) or citations from higher to lower courts.

A third notable project was the SCALIR system, designed by Rose and Belew.[255] It was built upon the connectionist model of information retrieval,[256] in which a network is constructed between terms and other symbols found in the corpus, combined with machine learning techniques. In many aspects it resembles a neural network.[257] The interesting thing about SCALIR in this perspective was that it used both inferred connections between nodes (which is common in connectionist models) but also typed connections representing actual legal citations, be them from statute-to-statute or case-to-case.

[252]Marx: Citation Networks in the Law (see n. 157).

[253]For example, the EURLEX service offers this possibility for all documents in the system through the "Select all documents mentioning this document"

[254]Tapper: The use of citation vectors for legal information retrieval (see n. 208) and a more complete report in Tapper: An experiment in the use of citation vectors in the area of legal data (see n. 158)

[255]Rose/Belew: A connectionist and symbolic hybrid for improving legal research (see n. 40).

[256]Described in Turtle: Text Retrieval in the Legal World (see n. 17), p. 34

[257]More properly called articifial neural networks, these are composed of a large number of units ("neurons") connected by input and output links that have a associated mutable weight. Each unit performs local computations based on information from input links, and distribute the result through its output links. The exact configuration of the network and how computation is performed is not explicitly designed, but evolved through training the network with a learning algorithm. See Stuart Russel/Peter Norvig: Artificial Intelligence - A Modern Approach, New Jersey 1995, p. 567

Unfortunately, these attempts were not followed by many other experiments during the 1980s and 1990s. It was sometimes remarked that legal IR systems should make better use of citations,[258] and that they were underexploited in retrieval systems.[259] It is only in the last few years that a number of promising new approaches to utilizing citation networks in legal IR have begun to show up. These projects are described in more detail in sec. 4.3

3.3 Formalizing a function for jurisprudential relevance

One way of identifying landmark cases is to observe which cases gets referenced the most often by peer cases relating to the same topic.[260]

Popularity might not be the same as relevance, but as the success of link analysis ranking in general IR show, it might be a useful *indicator* of relevance. Regarding case law, one must ask the question "If no one ever references a legal case, what are the chances of it being relevant". Especially in areas with a lot of precedents, the fact that a case has gone uncited for years may indicate that it does not contain useful information, even if it may appear to be on point.

If we do assume that popularity is a useful indicator of relevance, we can design a ranking function for legal IR systems that take into account the relationship between documents.

This function can be expressed in several forms. We therefore define some parameters for the function.

The first parameter is the basic link analysis ranking algorithm used. This algorithm should be run on the graph or set of graphs determined by the other two parameters. In the prototype system, three such algorithms are available: InDegree, PageRank and HITS. As HITS produce two values for each document (its hub value and its authority value), we select the authority value. This follows the suggestions from Fowler and Jeon's examination of the US Supreme Court citation network.[261]

The second parameter is the subset of the total citation network that we should use. The general rule is that to construct the citation network for a particular TFEU article, we only count citations from documents that cite that particular article. We can further restrict the graph by only counting citation to cases which in turn also cite that particular article.

The third parameter is how to account for the fact that some cases may

[258] Dabney: The Curse of Thamus: An analysis of Full-Text Legal Document Retrieval (see n. 7), p. 40.

[259] Turtle: Text Retrieval in the Legal World (see n. 17), p. 48.

[260] David James Miller/Harry R. Silver/Andrew L. Freisthler: Landmark Case Identification System and Method, US Patent 2006/0041608, Feb. 2006, at 0009.

[261] Fowler/Jeon: The authority of Supreme Court precedent (see n. 217).

only have had a few years to gain authority, while others that in earlier years may have gained large amounts of authority but none in recent years (age compensation). We can either use no compensation and only consider scores as they are determined by the current citation network. Or we can use a year-based averaging of scores by calculating authority scores from the citation network graph of all cases that existed in 1955, then the graph of all cases that existed in 1956, and so on, and then divide the sum of those authority scores with the age in years of each case.