Relevance Feedback Search Based on Automatic Annotation and Classification of Texts

Rafael Leal
HELDIG Centre for Digital Humanities, University of Helsinki, Finland

Joonas Kesäniemi
Semantic Computing Research Group (SeCo), Aalto University, Finland

Mikko Koho
HELDIG Centre for Digital Humanities, University of Helsinki, Finland
Semantic Computing Research Group (SeCo), Aalto University, Finland

Eero Hyvönen
Semantic Computing Research Group (SeCo), Aalto University, Finland
HELDIG Centre for Digital Humanities, University of Helsinki, Finland

Abstract

The idea behind Relevance Feedback Search (RFBS) is to build search queries as an iterative and interactive process in which they are gradually refined based on the results of the previous search round. This can be helpful in situations where the end user cannot easily formulate their information needs at the outset as a well-focused query, or more generally as a way to filter and focus search results. This paper concerns (1) a framework that integrates keyword extraction and unsupervised classification into the RFBS paradigm and (2) the application of this framework to the legal domain as a use case. We focus on the Natural Language Processing (NLP) methods underlying the framework and application, where an automatic annotation tool is used for extracting document keywords as ontology concepts, which are then transformed into word embeddings to form vectorial representations of the texts. An unsupervised classification system that employs similar techniques is also used in order to classify the documents into broad thematic classes. This classification functionality is evaluated using two different datasets. As the use case, we describe an application perspective in the semantic portal *LawSampo – Finnish Legislation and Case Law on the Semantic Web*. This online demonstrator uses a dataset of 82 145 sections in 3725 statutes of Finnish legislation and another dataset that comprises 13 470 court decisions.

2012 ACM Subject Classification Computing methodologies → Information extraction; Applied computing → Document searching; Information systems → Clustering and classification

Keywords and phrases relevance feedback, keyword extraction, zero-shot text classification, word embeddings, LawSampo

Digital Object Identifier 10.4230/OASIcs.LDK.2021.18

1 Introduction

In many search situations, the information need of the user cannot be formulated precisely. The search query must then be gradually refined and the results re-evaluated in a series of successive rounds in order to achieve a satisfactory outcome. This paper describes language technology algorithms used in the application of this kind of iterative and interactive method, the Relevance Feedback Search (RFBS) paradigm [1, Ch. 5], to the search and exploration of textual documents. We outline a search system that integrates keyword extraction and unsupervised categorization of documents into RFBS based on pre-trained models and algorithms. We also present a case study with an implementation of this framework to the legal domain as part of the *LawSampo – Finnish Legislation and Case Law on the Semantic Web*[1] [6] system.

[1] Project homepage: `https://seco.cs.aalto.fi/projects/lakisampo/en/`

Algorithm 1 Proposed relevance feedback search.

```
   Result: Result set RS in documents D satisfying the user's information need
   /* Initialization                                                    */
 1 AQ := ε ; // Active free-form text query (initially the empty string)
 2 AK := ∅ ; // Active Keywords set (initially empty)
 3 AC := ∅ ; // Active Categories set (initially empty)
 4 RS := {d_0, ..., d_n} ; // Result Set (initially contains the whole document
     domain)
   /* RFBS loop                                                         */
 5 while RS is not a satisfying result do
 6 |   SK := KeywordsOf(RS) ; // Suggested keywords based on RS
 7 |   SC := CategoriesOf(RS) ; // Suggested categories based on RS
 8 |   AQ := ModifyQ(AQ) ; // User optionally modifies AQ
 9 |   AK := ModifyK(AK, KW) ; // User optionally modifies AK based on SK
10 |   AC := ModifyC(AC, SC) ; // User optionally modifies AC based on SC
11 |   RS := Search(AQ, AK, AC);
12 end
```

The proposed RFBS process is outlined in Algorithm 1. The function $Search(AQ, AK, AC)$ in line 11 is used to search documents based on the search query AQ, active keywords AK, and active categories AC. The idea is that the categories are used for setting larger thematic contexts for the search, which can then be refined using keywords. For example, categories may refer to different phases or situations during the life of the end users, such as childhood or golden age [18], or societal contexts, such as health or environmental issues. In each iteration of the RFBS loop (lines 5–12), the system computes a set of new categories and keywords based on the search results RS as suggestions for the user to consider. The functions *ModifyQ*, *ModifyK* and *ModifyC* in lines 8-10 allow the user to make optional modifications for the next search round. In this way, the process is expected to converge gradually towards more and more satisfying results in RS.

A novel idea in the proposed approach is to combine implicit and explicit feedback methods [15] by using topical classification of the documents and keyword concepts extracted from the search results. User feedback on the topics and keywords is used to generate new search queries, thus guiding the iterative search process.

In order to avoid the challenges of traditional text-based search methods, for example morphological variation of words in highly inflectional languages such as Finnish, and semantic difficulties such as synonymy and polysemy, the presented system works on a semantic level. This is done based on ontological concepts and themes extracted automatically from the texts, which are processed via a word embedding algorithm. A zero-shot classification sub-system based on the same approach is responsible for categorizing the documents.

The implementation of the system uses Finnish legal documents, in particular sections of the law and court decisions, with the aim to help users to find jurisprudence related to their life situations. For the data, we use documents from the Semantic Finlex data service[2] [11], refined further for the LawSampo system and data service [6]. This search prototype constitutes one application perspective[3] of the LawSampo system. It is designed to work with Finnish language content, but the methods presented here can be applied to documents in any language, if similar ontologies and language processing tools are available.

[2] `https://data.finlex.fi/`
[3] In Sampo portals, the different ways for accessing the data are called "perspectives".

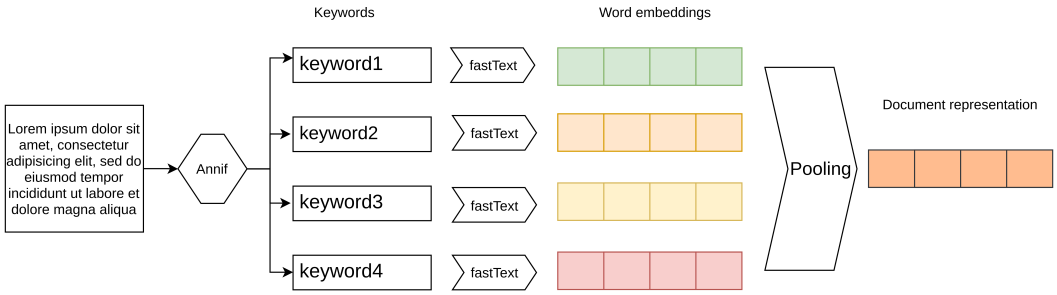

Figure 1 Simplified visual overview of the document representation building algorithm. The *document matrix* D collects the representations for all the documents in the dataset.

In the next sections, the methods and software needed for knowledge extraction of thematic categories and keywords from the texts are described, alongside some evaluation results. Subsequently, the data underlying our case study is explained, as well as the RFBS application in LawSampo. Finally, the contributions of our work are discussed alongside related works, and directions for further research are suggested.

2 Framework for Knowledge Extraction and Search

All documents in the target dataset are semantically annotated in two ways: 1) by extracting keyword concepts based on a keyword ontology and 2) by classifying the documents into a set of larger thematic categories. The search framework relies on three main components: the internal representation of the documents, the classifier, and the search algorithm. These will be explained in more detail in the following subsections.

2.1 Document Representation

The core of this system lies in its representations of the documents, which are built using a three-step approach: 1) representative keywords are extracted from the text, then 2) transformed into their respective word embeddings, and finally 3) combined via mean pooling to form the document representation vector. This process is visualized in Figure 1.

Keyword extraction is performed via Annif [4] [21], a subject indexing tool developed by the National Library of Finland. It is capable of using different algorithms to return suggested keywords and their respective weights for a given input text. Annif developers also provide various models, with different combinations of algorithms, on the Finto REST API[5]. They are all trained on bibliographical metadata from the works found on the Finna portal[6], which publishes information about objects in Finnish archives, museums and libraries, in a vein similar to Europeana. Since the training data is labelled with terms from the General Finnish Ontology (YSO)[7] [17] and its sister ontology YSO places (YSO-paikat), the API returns keywords identified by unique YSO URIs. This also means that the keywords obtained are already lemmatized. YSO contains a total of 31 205 main concept labels and an additional 7807 are available in YSO places.

[4] https://annif.org/
[5] https://ai.finto.fi/
[6] https://www.finna.fi/
[7] https://finto.fi/yso/en/

Our present system uses the *yso-fi* pre-trained model on the Finto REST API. This model is an ensemble, with equal weights, of three algorithms:

- a modified version of **TF-IDF**, which pre-calculates TF-IDF vectors for each *keyword* and compares them to similarly built document vectors to find the best matches [21, p. 8];
- **Maui**, an n-gram-based keyword extractor [9];
- **Parabel**, an extreme multi-label classification algorithm [13].

The first two directly match salient terms to a vocabulary, while the last is also able to find indirect correlations between words [21]. This mix provides results that are not only grounded on the text of the documents but also able to extrapolate their specific wording. A limit of 100 keywords per text and a minimum weight threshold of 0.01 are used, which offer a wider range of results than the defaults (respectively 10 and 0) and proved effective in the classification task [8]. The keyword weights obtained from Annif are saved into the *keyword matrix* K, a document-term structure.

Once the texts are distilled into sets of representative keywords, our system uses fastText [3] to obtain word embeddings for each of them. This combination of algorithms improves the task of unsupervised categorization by avoiding creating representations at the sentence level. In [8], various combinations of text representation and embedding algorithms were tested, including transformer-based algorithms such as FinBERT and S-BERT/XLM-R. Nevertheless, it concluded that fastText, especially in partnership with Annif, produced the best results using this technique.

FastText improves on the word2vec embedding algorithm by breaking up the words into a bag of n-grams, each with their own vectorial representation. When building the final embedding, these components are also taken into account. This mechanism provides the means to build representations for words that are not in the training data by breaking them up into smaller, already-seen chunks. This is especially important for morphologically rich languages such as Finnish, which present unseen word forms more often than analytic languages.

Representations are calculated via the `get_sentence_vector` function from the fastText Python package, which averages the l2 norm of the respective representation vector for each word in the sentence. The system uses a pre-trained language model offered by the fastText developers, trained on the Wikipedia (2017 dump) and Common Crawl datasets [5]. It has a dimension of 300 and a vocabulary containing the 2 million most common word tokens in the training data – around 18K concept labels from YSO are nevertheless not among them.

The final representation for each document is obtained via mean pooling of their keyword embeddings. All document representations are then collected into a $D^{d\times 300}$ *document matrix*, where d is the total number of documents and 300 is the number of dimensions in the pre-trained fastText model.

2.2 Document Classification

A similar process is used to build a $C^{c\times 300}$ *category matrix*, where c is the total number of category labels. Although the keyword extraction algorithms are trained on categorical metadata, as explained in Section 2.1, the classification mechanism that employs it can be considered unsupervised, since it is built without any training.

Category labels must be provided by the end user, since at this point the system does not recognise broad topics automatically. The labels are pre-processed by stripping commas and conjunctions and used as input for the category label representation pipeline, illustrated in Figure 2.

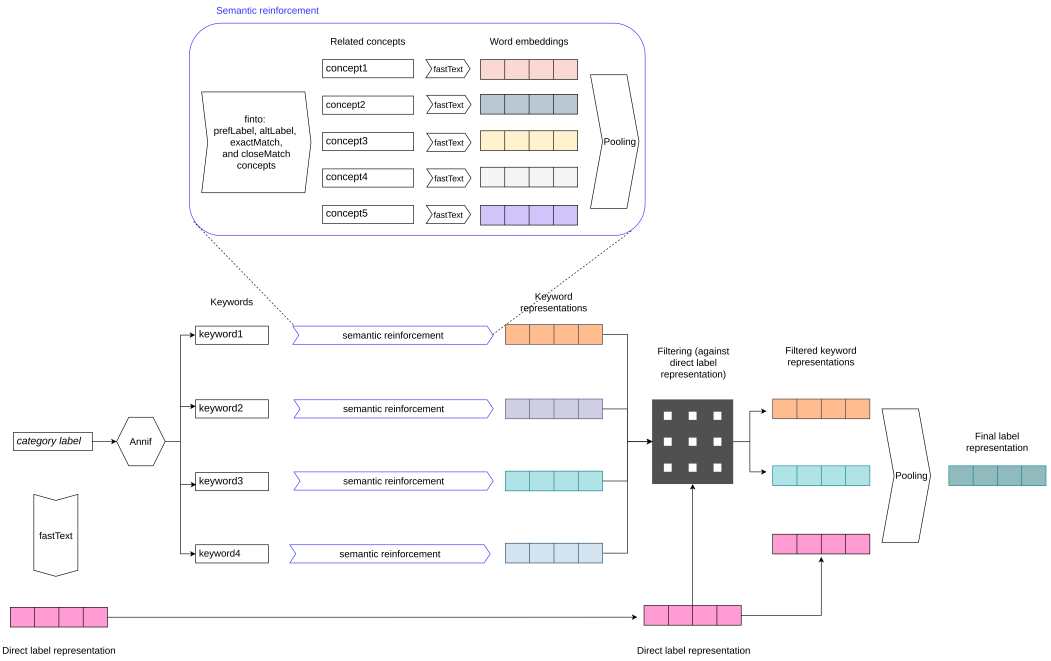

Figure 2 Simplified visual overview of how category label representations are built. First, related keywords obtained via Annif are semantically reinforced and transformed into vectors, then they are compared to an embedding of the original label (*direct label representation*). Those which score above a threshold are pooled together into the *final label representation*. The *category matrix C* contains the final label representations for all category labels.

Annif's results can be somewhat erratic when it comes to very short texts, and the category labels often contain a single word. Thus, filtering out unrelated terms from the set of keyword suggestions can thus produce better results. This is done via cosine similarity: a matrix containing word embeddings for the entire category label text as well as its individual words is built, and then compared to another matrix containing *semantically reinforced* embeddings for all the keywords suggested by Annif (this reinforcement technique is described below). If the maximum cosine similarity between a suggestion and any of the query vectors is under a certain threshold (0.25 is the default), the keyword is rejected. The word embeddings for the original label and the resulting keywords suggestions, pooled together, form the category representation vector.

Since Annif returns YSO concepts, their vectorial representation can be semantically reinforced. This simple technique is based on ontology relations: main and alternative labels (*prefLabel* and *altLabel*), exactly matching (*exactMatch*) and closely matching (*closeMatch*) entities linked to the target concept are fetched; the final representation is calculated as the average embedding of this expanded set of entities.

As an example, the concept *shares*, whose URI is `http://www.yso.fi/onto/koko/p12994`, returns a wealth of associated concepts: broader, narrower, related, exactly matching and closely matching concepts, as well as sections named "in other languages" and "entry terms", which are alternative labels for the concept. Out of those, the framework chooses only the following:

- ***prefLabel*** (the main label): *shares*
- ***altLabel*** (entry terms): *share, stocks*
- ***closeMatch*** (closely matching categories): *Stocks* (linked to the Library of Congress concept)

- ***exactMatch*** (exactly matching categories): *share, shares* (linked to other Finnish ontologies)

The associated terms are then collected in a set $S = \{shares, share, stocks\}$ and the final representation will be an average of the word embeddings for the terms in S. In this specific case, the procedure helps disambiguate this concept from another *shares* entry[8], which represents technical objects also known as *drill coulters* or *ploughshares*. However, this technique can help build better representations even without disambiguation, by pooling together similar concepts instead of relying on a single label.

The dot product between the *category matrix* C and the *document matrix* D builds a $V^{c \times d}$ *category-document matrix*, which stores the strength of each category-document pair.

2.3 Search

The search results obtained by our RFBS framework depends on four elements: text query, category, positive keywords, and negative keywords. They may work separately or in tandem. This algorithm corresponds to the *Search*() method in line 11 of Algorithm 1, in which both positive and negative keyword sets are represented by AK.

The free-form search query, in case it is given, is the first element to be resolved. It is transformed into embeddings in the standard way, via Annif→fastText. The results can be filtered as explained in section 2.2, however without semantic reinforcement in order to save computational time. The most similar documents are calculated via cosine similarity between the query representation and the *document matrix* D, and the results are collected in a result list.

Positive keywords and categories are executed next, in case they are set. First, scores for each document are calculated as the sum of their respective weights (from the keyword matrix K) for all the positive keywords in question. These scores are used either as weights for an already-existing result list or to create a result list from scratch. Category scores are simply the respective row of the *category-document matrix* V. Similarly to the previous step, these scores are used either as weights or to create a new result list in case no other search parameters are set. Finally, negative keywords are used to exclude all documents from the result list containing any of the negative keywords chosen.

Next, the system calculates keywords candidates and category candidates. These are equivalent to the methods *KeywordsOf*() and *CategoriesOf*() in lines 6 and 7 of Algorithm 1. Keywords are calculated by averaging the respective *keyword matrix* K scores for all documents in the result list. Additionally, category candidates are calculated by averaging the category scores for the relevant documents in the *category-document matrix* V.

3 Evaluation of the System: Current Status

In this section, we present an overview of the evaluation of the framework components, focusing on the classification system.

3.1 Classification

The classification component of this system has been evaluated with two different datasets. Neither is representative of the dataset in the Use Case, so they can be both considered baselines. The datasets are the following:

[8] `http://www.yso.fi/onto/koko/p48662`

- A custom Yle dataset, containing 5096 news articles from Yle (Yleisradio Oy), the Finnish Broadcasting Company[9], classified in 11 categories according to their main tag: *politiikka*, 'politics', *talous* 'economy', *kulttuuri* 'culture', *luonto* 'nature', *tiede* 'science', *terveys* 'health', *liikenne* 'transportation', *urheilu* 'sports', *sää* 'weather', *parisuhde* 'intimate relationships' and *rikokset* 'crimes'). This dataset is equivalent to *Yle 1* and *Yle 2* in [8] combined[10]. It has a mean length, in characters, of 2110.8 ± 1534.5.
- The Minilex dataset: a collection of 2567 legislation-related questions classified in 13 categories (*asunto, kiinteistö* 'housing, real estate', *immateriaalioikeus* 'intellectual property', *irtain omaisuus* 'chattel', *lainat, velat* 'loans, debts', *liikenne* 'transportation', *oikeudenkäynti* 'legal proceedings', *perheoikeus, perintöoikeus* 'family law, inheritance law', *rikokset* 'crimes', *sopimus, vahingonkorvaus* 'contracts, compensation', *työsuhde, virkasuhde* 'employment, public service', *ulosotto, konkurssi* 'debt recovery, bankruptcy', *vuokra* 'rent', *yritykset, yhteisöt* 'companies, organisations') from the legal services website Minilex[11]. As can be observed, there is semantic overlap among some of the categories in this dataset, such as "debt recovery, bankruptcy" / "loans, debts"; or "housing, real estate" / "rent" / "contracts, compensation"; or even "transportation" / "crimes". The mean length of his dataset is 447.3 ± 236.1 characters.

Three different measures were taken: the F_1 score, the Mean Reciprocal Rank (MRR), and the rank of the gold label among the predictions made by the classifier. The last metric springs from the fact that this system is used as a multi-label classifier, so not having the gold tag as the best prediction is not necessarily a consequential result.

The system fares better with the Yle dataset, with an F_1 score of 0.737, an MRR of 0.828 and a mean rank of 0.7 ('0' being the gold label, '1' the second prediction, etc; lower is thus better), while the Minilex dataset obtains an F_1 score of 0.56, an MRR of 0.697 and a mean rank of 1.557. Using semantically reinforced vectors gives a small boost to these numbers: Yle gets an F_1 of 0.742, an MRR of 0.832 and a mean rank of 0.67, and Minilex respectively 0.572, 0.705 and 1.496. Table 1 details the counts and cumulative distributions of the results.

These numbers show that the present classification method is capable of categorizing the gold labels as the top prediction in 57–74% of the cases and among the top 3 predictions in 80–90% of the cases, depending on the dataset used. This variation can possibly be attributed to a combination of the length of the documents and the quality of the category labels. This classifier tends to fare better with longer texts [8], so the Yle dataset, which contains texts that are around four times longer on average, has a starting advantage. Moreover, as discussed above, the Minilex labels are semantically more similar, which increases the probability of obtaining non-optimal predictions.

Legislative texts may present more difficulties for this system, since they contain a more peculiar and jargonic choice of words. Lack of familiarity with the words is a challenge for both the keyword-extracting and the word embedding algorithms, which may not recognize or represent them properly.

At any rate, the results presented here are not in line with state-of-the art supervised algorithms, which reach results above 95% and nearing 100% for their top predictions (cf. survey in [10]). However, training data in Finnish is hard to obtain, and without it a

[9] `https://yle.fi`
[10] Excluding one article, whose main tag has changed. This data is not redistributable.
[11] `https://www.minilex.fi`. Their terms of use allows for non-commercial usage of the data, but not for its redistribution.

Table 1 Count and cumulative distribution (CUMD) of results in the classification task for the Yle and Minilex datasets, in both their standard and reinforced versions.

	Yle				Minilex			
	Standard		Reinforced		Standard		Reinforced	
rank	COUNT	CUMD	COUNT	CUMD	COUNT	CUMD	COUNT	CUMD
0	3757	0.737	3779	0.742	1438	0.56	1469	0.572
1	560	0.847	568	0.853	406	0.718	388	0.723
2	269	0.9	264	0.905	179	0.788	187	0.796
3	147	0.929	151	0.934	102	0.828	104	0.837
4	115	0.951	96	0.953	90	0.863	78	0.867
5	91	0.969	104	0.974	59	0.886	64	0.892
6	68	0.983	57	0.985	86	0.919	86	0.926
7	38	0.99	30	0.991	61	0.943	62	0.95
8	25	0.995	22	0.995	56	0.965	44	0.967
9	19	0.999	18	0.999	39	0.98	38	0.982
10	7	1.0	7	1.0	32	0.993	28	0.993
11	-	-	-	-	18	1.0	18	1.0
12	-	-	-	-	1	1.0	1	1.0

supervised system cannot be built. The main advantages of the classification algorithm presented in this paper are its flexible nature – since it only requires a list of category labels in order to work – and its straightforward integration into the search framework, since it uses the same underlying technologies.

3.2 Document Representation

These results also show strong consistency on the part of Annif: it is capable of coherently assigning keywords to both documents and category labels, so that most documents can be correctly classified when transposed to a vector space of embeddings. FastText also demonstrates reliability performing this transposition from semantic meanings to vectorial representations.

A more decisive evaluation of this component is planned as part of our future work.

3.3 Search

No formal evaluation of the search system has been carried out so far. However, Section 4 contains tests and insights about the reliability and usability of this component.

4 Use Case: Search Engine for Finnish Legislation and Case Law

This section describes, via a concrete use case, how the RFBS has been adapted to the legal domain as part of the semantic portal LawSampo [6]. LawSampo is the first Sampo portal to add RFBS-based functionality to complement the search features of previous Sampos, which are mostly facet-based.

4.1 The Data

LawSampo contains data about Finnish legislation and case law as a harmonized Linked Data knowledge graph. This knowledge graph is based on Semantic Finlex data [11], filtered and transformed into a simpler data model. This was done with the aim of hiding the inherent complexity of legal documentation while keeping the data relevant to anyone interested in the topic. These data transformations were implemented as SPARQL CONSTRUCT queries, and the data is enriched with information about referenced EU legislation, as well as the generated annotations of subject keywords and category labels.

The set of category labels used in LawSampo's RFBS system is based on the Minilex dataset discussed in Section 3. However, in order to avoid the semantic overlap present among some of the categories and to expand the field of possible categories, they were refined with some input from experts at the Ministry of Justice of Finland. The resulting set contains 12 categories: *asuminen, kiinteistö* 'housing, real estate', *ihmisoikeudet, perusoikeudet* 'human rights, basic rights', *omaisuus, kaupankäynti, kuluttajansuoja* 'property, commerce, consumer protection', *julkishallinto, valtionhallinto* 'public administration', *rahoitus* 'finance', *verotus* 'taxation', *yritykset, yhteisöt, työelämä* 'business, organizations, working life', *liikenne, kuljetus* 'traffic', *perheoikeus, perintöoikeus* 'family law, inheritance', *rikosasiat, oikeudenkäynti* 'crime, legal proceedings', *koulutus* 'education', *ympäristö* 'environment'.

The consolidated legislation consists of 3725 statutes and their 82 145 sections, with each section consisting of the most current version of the full-text contents in Finnish. The case law dataset consists of 13 470 court decisions and their full-text contents.

4.2 LawSampo User Interface for RFBS

The LawSampo portal implements the RFBS Algorithm 1 in its "Contextual Search" application perspective[12]. LawSampo allows the user to initialize the RFBS system either by setting a free-form text query or by selecting one of the provided document categories from the list presented in Section 4.1. Figure 3 illustrates the user interface after the user has started a search via a text query. After the initial search, the application switches to an iterative mode corresponding to the while-do loop (lines 5-12) of Algorithm 1, where the search is fine-tuned by managing a set of active filters through categories and keywords suggested by the system. Since no category was selected in the initial phase in Figure 3, the user is presented with a top-three list of suggested categories (obtained in line 7 in the algorithm) in addition to 20 suggested keywords (line 6). In this implementation, the values used for the initial search cannot be changed during the iterative loop: if the initial filter (query or category) is removed, the iteration stops and the search returns to its empty initial state.

Table 2 shows a simple example of how, using the same query, "right of redemption", different categories affect the resulting documents. With the given query, it is not surprising that *Act on the Redemption of Immovable Property and Special Rights* is the most relevant hit. More interestingly, the *Water Act* becomes the second-most referenced document when the "human rights, basic rights" category is active, whereas choosing the "crime, legal proceedings" category surfaces the *Building Act*. Finally, the "property, commerce and consumer protection" category adds *Act on the Residential and Commercial Property Information System* to the returned documents.

[12] In the Sampo model, the user is provided with several independent but interlinked applications that use a shared underlying knowledge graph.

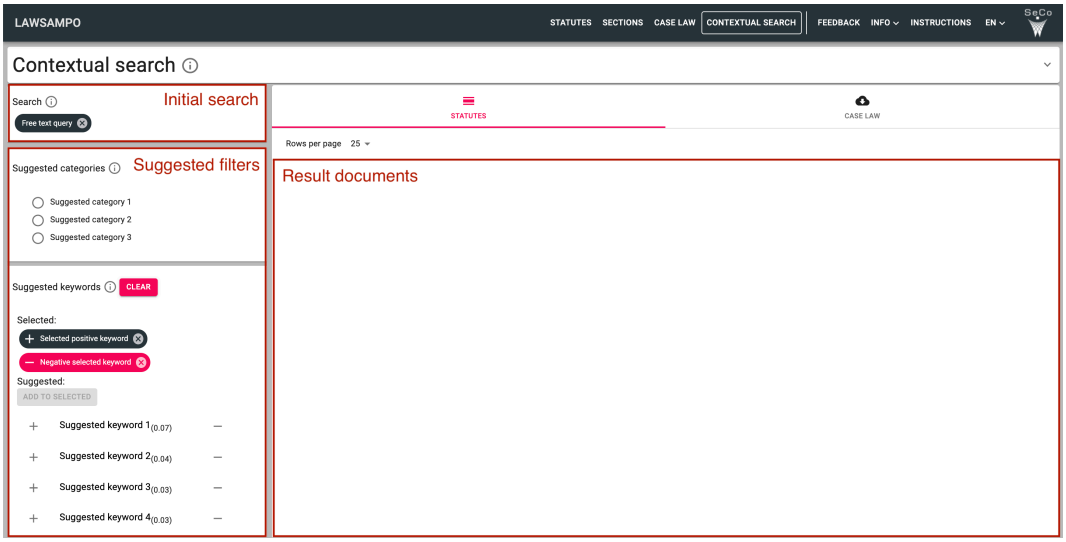

Figure 3 LawSampo's contextual search UI. The user has made an initial text query and selected one positive and one negative keyword. Next, the user can continue the search by using the suggested categories and keywords in order to modify the active filters. The number shown in subscript next to the keywords is a normalized relevance score.

Table 2 How does the selected category affect the resulting documents? This table shows statute results for the query *lunastusoikeus* 'right of redemption' with three different categories and a result size of five. The values represent the number of sections returned from each statute.

	crime, legal proceedings	**human rights, basic rights**	**property, commerce, consumer protection**
Water act	1	2	1
Building act	1		
Real Estate Formation Act	1		1
Act on the Residential and Commercial Property Information System			1
Act on the Redemption of Immoveable Property and Special Rights	2	3	2

Suggested keywords are shown with a relevance score calculated on the basis of the current list of resulting documents. Keywords can be added to the active filters (line 9 in Algorithm 1) in either positive or negative mode using the plus (+) or minus (−) buttons respectively. A negative selection excludes any documents containing the given keyword from the result list, as explained in Section 2.3.

The result list view can be toggled between statutes and case law documents as two parallel tabs. The results are updated whenever the user changes any of the active filters, i.e., textual query, category or selected positive and negative keywords. The documents returned are statute sections or case law abstracts. The user is given the possibility to skim through them and to follow links to the full documents. Since the statutes search works at the section level, the results can contain multiple documents from the same statute.

Table 3 Relevance feedback development during the first search iterations in the example scenario. *Iteration 0* is the initial search, which is done without any selected keyword filters. Positive and negative keywords are denoted with "+" and "−" characters respectively. The *Suggested keywords* row only shows a subset of the keywords suggested by LawSampo's contextual search. The italic typeface is used to mark keywords selected for the next iteration.

	Iteration 0	**Iteration 1**	**Iteration 2**	**Iteration 3**
Query	"expropriation lot"			
Selected keywords		+ redemption + right for redemption − alluvial land − compulsory auction	+ redemption + right for redemption − alluvial land − compulsory auction	+ compensation for redemption − alluvial land − compulsory auction
Document type	Statutes	Statutes	Case law	Statutes
Suggested keywords	real property, *redemption*, *right for redemption* cadastral procedures, land surveying, *alluvial land*, *compulsory auction*	savings banks, railways, limited companies, shares, town planning, land use policy, company law	roads, construction, municipalities, land use planning, land acquisition, land use, *compensation for redemption*	indemnities, prices, value (properties), interest (economics), owners

4.3 Search Example Scenario

This section presents as an example of how, using LawSampo's "Contextual Search", the following information need can be satisfied:

> *I've been thinking about making a huge renovation to our house. However, I'm worried that because our house is in such a good area, the city might expropriate the lot. If that happens, what kind of payout could I be looking at?*

Let us begin the search with a simple query based on the information need described above: *pakkolunastus tontti* 'expropriation lot'. The query is executed against the statutes by default and, for this example, the maximum number of results is set to 10. A summary of the RFBS process for each search iteration can be found in Table 3.

The document list resulting from the first iteration (RS_1) does not look very promising: the only vaguely relevant document is *Erämaalaki* 'Wilderness Act', which describes how the state can expropriate land in wilderness areas in order to build roads. The result list also contains multiple non-relevant documents related to forced auction and water systems. For the next iteration, we add from the suggested keywords *lunastus* 'redemption' and *lunastusoikeus* 'right of redemption' as positive keywords and *vesijättömaa* 'alluvial land' and *pakkohuutokauppa* 'compulsory auction' as negative ones to our set of active filters (AK in Algorithm 1).

The second set of results (RS_2) is already more useful. There are two more references to expropriation related to roads and railroads that can be considered somewhat relevant, but also a match to the highly useful *Land use and Building Act*. The results also indicate that

the positive keyword filters might not work as intended, since the results include documents related to *limited companies*, which deal with the wrong kind of "redemption" with respect to our information need. The results even contain a section from the *Saving Back Act*, most likely due to the use of the term *redemption* in the document.

We can test our intuition about the keyword "redemption" by switching over to the case law view to verify if the results are similar: the case law results (RS_3) with the same set of active filters are indeed consistent with the results in the statute view, with a couple of only indirectly relevant documents. However, the suggested keywords (AK_3) now include *lunastuskorvaukset* 'compensation for redemption', which matches our information need perfectly.

Finally, let us replace the keywords "redemption" and "right of redemption" with "compensation from redemption" and swap back to the statutes view to retrieve one more result list (RS_4). This time, all returned documents can be considered useful, including three references to a document titled *Act on the Redemption of Immovable Property and Special Rights*.

5 Discussion

This section overviews earlier related research, summarizes the contributions made by this paper, and outlines paths for future research.

5.1 Related Work

Various methods exist for relevance feedback search [1, 15]. Teevan et al. [23] enrich web search with relevance feedback based on a constructed user profile. Peltonen et al. [12] combine visual intent modeling with exploratory relevance feedback search. Tang et al. [22] used topic modeling in academic literature search. Song et al. [20] employed topic modeling with relevance search, based on implicit feedback from the topics of the user web search history. In [7], RFBS is combined with topic modelling [2]. However, the method of combining automatic document classification and keyword extractions with relevance feedback search, as described in this paper, is novel.

Regarding zero-shot classification methods, most of them work by training a classifier and then adapting it to a new set of categories [4, 14, 24], while [26] also integrates a knowledge graph into their algorithm. Among unsupervised classification models, [16] use skip-gram word embeddings to calculate the semantic similarity between a label and the given documents, which is also the basis of our work. In contrast, [25] treats zero-shot as an entailment problem.

5.2 Contributions

This paper argued for using a combination of topical classification and ontological keywords as a semantic basis for RFBS when exploring textual documents from complex domains, such as legislation and case law. A method for accomplishing this was presented, as well as an implementation of it for testing and evaluating purposes.

The content annotation results shown in Section 3 indicate that the proposed classification system, despite its unsupervised nature, is capable of classifying documents correctly 74% of the time (or 90% within the first three predictions) when the classes are semantically non-overlapping and the texts are long enough.

Section 4 illustrated how the RFBS algorithm suggested in this paper can be used in practice, demonstrating how LawSampo's Contextual Searcher perspective can be used to navigate documents from a semantically complex domain successfully in an iterative fashion.

Automatically suggested keywords mitigate the burden on the user of coming up with suitable queries and can provide valuable feedback even when the user selects non-optimal keywords. We have not yet performed a more general evaluation of the search functionality besides testing the system in selected individual problems. Nevertheless, the experiments presented in this paper suggest that a combination of ontological keyword annotations and topical classifications with word embeddings can create a useful semantic basis for the RFBS paradigm when searching and exploring textual legal documents.

5.3 Future Work

More research will be done in order to improve the vectorial representation of the documents. As it stands, these representations are entirely based on each document's set of keywords, which add depth (by emphasizing these keywords) but subtract breadth (other details of the text) from them. We plan to pursue three main lines of research in the future: one line aims at improving the set of representative keywords by both filtering out unrelated suggestions and adding new ones via ontology relations, named entity recognition and other keyword extraction algorithms, provided they can be integrated into the system. The second line aims at investigating alternative representation vectors partly based on whole-text embeddings. A third line of research consists in the automatic identification of major topics in the data as an alternative to the user-provided category list: this can possibly be accomplished by capitalizing on existing clustering methods and ontological relations among the keywords.

LawSampo is the first portal in the Sampo series of systems[13] to take advantage of this search functionality based on relevance feedback. Similar methods are planned as part of the upcoming new Sampos as well. These include especially the ParliamentSampo system[14], which incorporates over 900 000 parliamentary debate speeches [19] from the Parliament of Finland (1907–2021), documents that are related to the legislative texts found in LawSampo.

References

1 R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval (2nd Ed.)*. Addison-Wesley Longman Publishing Co., Inc., 2011.

2 David M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, 2012. doi:10.1145/2133806.2133826.

3 Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146, December 2017. doi:10.1162/tacl_a_00051.

4 Yann N. Dauphin, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. Zero-Shot Learning for Semantic Utterance Classification. *ICLR 2014*, 2014. arXiv:1401.0509.

5 Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

6 Eero Hyvönen, Minna Tamper, Arttu Oksanen, Esko Ikkala, Sami Sarsa, Jouni Tuominen, and Aki Hietanen. LawSampo: A semantic portal on a linked open data service for finnish legislation and case law. In *The Semantic Web: ESWC 2020 Satellite Events. Revised Selected Papers*, pages 110–114. Springer–Verlag, 2019.

7 Mikko Koho, Erkki Heino, Arttu Oksanen, and Eero Hyvönen. Toffee - semantic media search using topic modeling and relevance feedback. In *Proceedings of the ISWC 2018 Posters &*

[13] https://seco.cs.aalto.fi/applications/sampo/
[14] https://seco.cs.aalto.fi/projects/semparl/en/

Demonstrations, Industry and Blue Sky Ideas Tracks. CEUR Workshop Proceedings, October 2018. Vol 2180. URL: http://ceur-ws.org/Vol-2180/.

8 Rafael Leal. Unsupervised zero-shot classification of Finnish documents using pre-trained language models. Master's thesis, University of Helsinki, Department of Digital Humanities, 2020. URL: http://urn.fi/URN:NBN:fi:hulib-202012155147.

9 Olena Medelyan. *Human-Competitive Automatic Topic Indexing*. Thesis, The University of Waikato, 2009. URL: https://researchcommons.waikato.ac.nz/handle/10289/3513.

10 Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep learning–based text classification: A comprehensive review. *ACM Computing Surveys (CSUR)*, 54(3):1–40, 2021. doi:10.1145/3439726.

11 Arttu Oksanen, Jouni Tuominen, Eetu Mäkelä, Minna Tamper, Aki Hietanen, and Eero Hyvönen. Semantic Finlex: Transforming, publishing, and using finnish legislation and case law as linked open data on the web. In G. Peruginelli and S. Faro, editors, *Knowledge of the Law in the Big Data Age*, volume 317 of *Frontiers in Artificial Intelligence and Applications*, pages 212–228. IOS Press, 2019. ISBN 978-1-61499-984-3 (print); ISBN 978-1-61499-985-0 (online). URL: http://doi.org/10.3233/FAIA190023.

12 Jaakko Peltonen, Jonathan Strahl, and Patrik Floréen. Negative relevance feedback for exploratory search with visual interactive intent modeling. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pages 149–159. ACM, 2017.

13 Yashoteja Prabhu, Anil Kag, Shrutendra Harsola, Rahul Agrawal, and Manik Varma. Parabel: Partitioned Label Trees for Extreme Classification with Application to Dynamic Search Advertising. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*, pages 993–1002. ACM Press, 2018. doi:10.1145/3178876.3185998.

14 Anthony Rios and Ramakanth Kavuluru. Few-Shot and Zero-Shot Multi-Label Learning for Structured Label Spaces. *EMNLP*, 2018. doi:10.18653/v1/D18-1352.

15 Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288, 1990.

16 Prateek Veeranna Sappadla, Jinseok Nam, Eneldo Loza Mencía, and Johannes Fürnkranz. Using semantic similarity for multi-label zero-shot classification of text documents. In *ESANN*, 2016.

17 Katri Seppälä and Eero Hyvönen. Asiasanaston muuttaminen ontologiaksi. Yleinen suomalainen ontologia esimerkkinä FinnONTO-hankkeen mallista (Changing a keyword thesaurus into an ontology. General Finnish Ontology as an example of the FinnONTO model). Technical report, National Library, Plans, Reports, Guides, March 2014. URL: https://www.doria.fi/handle/10024/96825.

18 Teemu Sidoroff and Eero Hyvönen. Semantic e-goverment portals - a case study. In *Proceedings of the ISWC-2005 Workshop Semantic Web Case Studies and Best Practices for eBusiness SWCASE05*, 2005. URL: https://seco.cs.aalto.fi/publications/2005/sidoroff-hyvonen-semantic-e-government-2005.pdf.

19 Laura Sinikallio, Senka Drobac, Minna Tamper, Rafael Leal, Mikko Koho, Jouni Tuominen, Matti La Mela, and Eero Hyvönen. Plenary debates of the Parliament of Finland as linked open data and in Parla-CLARIN markup, March 2021. Accepted, LDK 2021.

20 Wei Song, Yu Zhang, Ting Liu, and Sheng Li. Bridging topic modeling and personalized search. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1167–1175. Association for Computational Linguistics, 2010.

21 Osma Suominen. Annif: DIY automated subject indexing using multiple algorithms. *LIBER Quarterly*, 29(1):1–25, July 2019. doi:10.18352/lq.10285.

22 Jie Tang, Ruoming Jin, and Jing Zhang. A topic modeling approach and its integration into the random walk framework for academic search. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 1055–1060. IEEE, 2008.

23 Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In *Proc. of the 28th Annual International ACM SIGIR Conference*, SIGIR '05, pages 449–456. ACM, 2005.

24 Zhiquan Ye, Yuxia Geng, Jiaoyan Chen, Jingmin Chen, Xiaoxiao Xu, SuHang Zheng, Feng Wang, Jun Zhang, and Huajun Chen. Zero-shot Text Classification via Reinforced Self-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3014–3024. Association for Computational Linguistics, 2020. doi:10.18653/v1/2020.acl-main.272.

25 Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923. Association for Computational Linguistics, 2019. doi:10.18653/v1/D19-1404.

26 Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. Integrating Semantic Knowledge to Tackle Zero-shot Text Classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1031–1040. Association for Computational Linguistics, 2019. doi:10.18653/v1/N19-1108.