Automatic Classification Using DDC on the Swedish Union Catalogue

Koraljka Golub[1][0000-0003-4169-4777], Johan Hagelbäck[2][0000-0002-8591-1035] and Anders Ardö (emeritus)[3]

[1] Department of Cultural Sciences, Faculty of Arts and Humanities, Linnaeus University, Växjö, Sweden
koraljka.golub@lnu.se
[2] Department of Computer Science and Media Technology, Faculty of Technology, Linnaeus University, Kalmar, Sweden
johan.hagelback@lnu.se
[3] Department of Electrical and Information Technology, Lund University, Lund, Sweden
anders.ardo@gmail.com

Abstract. With more and more digital collections of various information resources becoming available, also increasing is the challenge of assigning subject index terms and classes from quality knowledge organization systems. While the ultimate purpose is to understand the value of automatically produced Dewey Decimal Classification (DDC) classes for Swedish digital collections, the paper aims to evaluate the performance of two machine learning algorithms for Swedish catalogue records from the Swedish union catalogue (LIBRIS). The algorithms are tested on the top three hierarchical levels of the DDC. Based on a data set of 143,838 records, evaluation shows that Support Vector Machine with linear kernel outperforms Multinomial Naïve Bayes algorithm. Also, using keywords or combining titles and keywords gives better results than using only titles as input. The class imbalance where many DDC classes only have few records greatly affects classification performance: 81.37% accuracy on the training set is achieved when at least 1,000 records per class are available, and 66.13% when few records on which to train are available. Proposed future research involves an exploration of the intellectual effort put into creating the DDC to further improve the algorithm performance as commonly applied in string matching, and to test the best approach on new digital collections that do not have DDC assigned.

Keywords: LIBRIS, Dewey Decimal Classification, automatic classification, machine learning, Support Vector Machine, Multinomial Naïve Bayes, subject access.

1 Introduction

Subject searching (searching by topic or theme) is the most common and at the same time the most challenging type of searching in library catalogs and related quality information services, compared to, for example, a known-title or a known-author search.

Subject index terms taken from standardized knowledge organization systems (KOS), like classification systems and subject headings systems, provide numerous benefits compared to free-text indexing of commercial search engines: consistency through uniformity in term format and the assignment of terms, provision of semantic relationships among terms, support of browsing by provision of consistent and clear hierarchies (for a detailed overview, see, for example, [1]). However, controlled subject index terms are expensive to produce manually and there is a huge challenge facing library catalogs and digital collections of various types: how to provide high quality subject metadata for increasing numbers of digital information at reasonable costs. (Semi)-automatic subject classification and indexing represent some potential solutions to retain the established objectives of library information systems.

With the ultimate purpose of establishing the value of automatically produced classes for Swedish digital collections, the paper aims to develop and evaluate automatic subject classification for Swedish textual resources from the Swedish union catalogue (LIBRIS[1]). Based on a data set of 143,756 records catalogue records, a machine learning approach was chosen and evaluated. Multinomial Naïve Bayes (NB) and Support Vector Machine with linear kernel (SVM) algorithms were applied.

The paper is structured as follows: next section (2 Background) sets out the rationale for the study and discusses challenges surrounding automatic subject indexing and classification when applied in quality information systems; in Methodology the data collection, two algorithms and evaluation are described; the Results section reports on major outcomes; in Concluding remarks section a brief discussion of the impact and proposed future research are given.

2 Background

Subject searching is a common type of searching in library catalogs [2-3] and discovery services [4]. However, in comparison to known-item searching (finding an information object whose title, author etc. is known beforehand) searching by subject is much more challenging. This is due to difficulties such as ambiguities of the natural language and poor query formulation, which can be due to lack of knowledge of the subject matter at hand and of information searching. In order to alleviate these problems, library catalogues and related information retrieval systems (could) employ:

1. Hierarchical browsing of classification schemes and other controlled vocabularies with hierarchical structures, which help further the user's understanding of the information need and provide support to formulate the query more accurately;
2. Controlled subject terms from vocabularies such as subject headings systems, thesauri and classification systems, to help the user to, for example, choose a more specific concept to increase precision, a broader concept or related concepts to increase

[1] http://libris.kb.se

recall, to disambiguate homonyms, or to find which term is best used to name a concept.

The Swedish National Library recently adopted the Dewey Decimal Classification (DDC) to be used as a new national classification system [5], replacing SAB (Klassifikationssystem för svenska bibliotek) used earlier since 1921. However, cataloguing with a major classification system, such as DDC, is resource intensive. While fully automatic solutions are not currently feasible, semi-automated solutions can offer considerable benefit, both in assisting the workflow of expert cataloguers and in encouraging wider use of controlled indexing by authors and other users. Although some software vendors and experimental researchers claim to entirely replace manual indexing in certain subject areas [6], others recognize the need for both manual (human) and computer-assisted indexing, each with its (dis)advantages [7-8]. Reported examples of operational information systems include NASA's machine- aided indexing which was shown to increase production and improve indexing quality [9]; and the Medical Text Indexer at the US National Library of Medicine, which by 2008 was consulted by indexers in about 40% of indexing throughput [10].

However, hard evidence on the success of automatic indexing tools in operating information environments, is scarce; research is usually conducted in laboratory conditions, excluding the complexities of real-life systems and situations. The practical value of automatic indexing tools is largely unknown due to problematic evaluation approaches. Having reviewed a large number of automatic indexing studies, Lancaster concluded that the research comparing automatic versus manual indexing is "seriously flawed" [1] (p. 334). One common evaluation approach is testing the quality of retrieval based on the assigned index terms. But retrieval testing is fraught with problems; the results depend on many factors, so retrieval testing cannot isolate the quality of the index terms. Another approach is to measure indexing quality directly. One method of doing so is to compare automatically assigned metadata terms against existing human-assigned terms or classes of the document collection used (as a 'gold standard'), but this method also has problems. When indexing, people make errors, such as related to exhaustivity (too many or too few subjects assigned) or specificity (usually because the assigned subject is not the most specific available); they may omit important subjects, or assign an obviously incorrect subject. In addition, it has been reported that different people, whether users or professional subject indexers, assign different subjects to the same document. For a more detailed discussion on these challenges and proposed approach, see [11].

Research related to automated subject indexing or classification is divided between three major areas: document clustering, text categorization and document classification [12-13]. In document clustering, both clusters (classes) into which documents are classified and, to a limited degree, relationships between them, are produced automatically. Labelling the clusters is a major research problem, with relationships between them, such as those of equivalence, related-term and hierarchical relationships, being even more difficult to automatically derive [14]. In addition, "[a]utomatically-derived structures often result in heterogeneous criteria for class membership and can be difficult to understand" [15] (p. 146). Also, clusters' labels, and the relationships between them,

change as new documents are added to the collection; unstable class names and relationships are user-unfriendly in information retrieval systems, especially when used for subject browsing. Related to this is keyword indexing whereby topics of a document are identified and represented by words taken from the document itself (also referred to as derived indexing).

Text categorization (machine learning) is often employed for automatic classification of free text. Here characteristics of subject classes, into which documents are to be classified, are learnt from documents with manually assigned classes (a training set). However, the problem of inadequate training sets for the varied and non-uniform hierarchies of the DDC has been recognized. [16] argues that DDC's deep and detailed hierarchies can lead to data sparseness and thus skewed distribution in supervised machine learning approaches. [17] classified scientific documents to the first three levels of DDC from the Bielefeld Academic Search Engine. They found an "asymmetric distribution of documents across the hierarchical structure of the DDC taxonomy and issues of data sparseness", leading to a lack of interoperability that was problematic.

In the document classification approach, string matching is conducted between a controlled vocabulary and the text of documents to be classified [12-13]. A major advantage of this approach is that it does not require training documents, while still maintaining a pre-defined structure of the controlled vocabulary at hand. If using a well-developed classification scheme, it will also be suitable for subject browsing in information retrieval systems. Apart from improved information retrieval, another motivation to apply controlled vocabularies in automated classification is to re-use the intellectual effort that has gone into creating such a controlled vocabulary. It can be employed with vocabularies containing uneven hierarchies or sparse distribution across a given collection. It lends itself to a recommender system implementation since the structure of a prominent classification scheme, such as the DDC, will be familiar to trained human indexers.

Automatic document classification based on DDC remains challenging. In early work, OCLC reported on experiments in the Scorpion project to automatically classify DDC's own concept definitions with DDC [18]. The matching was based on captions. In more recent work, relative index terms from DDC were also incorporated [19]; the aim was to investigate automatic generation of DDC subject metadata from English language digital libraries in the UK and USA. The algorithm approximates the practice of a human cataloguer, first identifying candidate DDC hierarchies via the relative index table and then selecting the most appropriate hierarchical context for the main subject. Using measure called mean reciprocal rank, calculated as 1 divided by the ranked position of the first relevant result, they achieved 0.7 for top 2 levels of DDC and 0.5 for top 3 levels. They considered the results competitive and promising for a recommender system. [20] and [21] use a different controlled vocabulary and also report on competitive results.

3 Methodology

3.1 Dewey Decimal Classification (DDC)

The DDC was named after its conceiver Melvil Dewey; its first edition was published in 1876. Today the DDC is the most widely used classification system in the world: it has been translated to over 30 languages and is used by libraries in more than 130 countries.

The DDC covers the entire world of knowledge. Basic classes are organized by disciplines or fields of study. At the top level there are 10 main classes each of which is further divided into 10 divisions; each division is further subdivided into 10 sections. As a result, the DDC is hierarchical, and well serves purposes of hierarchical browsing. Each class is represented using a unique combination of Arabic numerals which are the same in all languages, providing the potential for cross lingual integrated search services.

The first digit in the class number represents the main class, the second digit indicates the division, and the third digit the section. For example: 500 stands for sciences, 530 for physics, 532 for fluid mechanics. The third digit in a class number is followed by a decimal point used as a psychological pause since after that the division by 10 continues to a number of other more specific degrees of classification, as needed.

The DDC research permit, the Swedish language version, edition 23, was obtained by the research team from OCLC in 2017. The file received was in MARCXML format comprising over 128 MB. For ease of application, relevant data were extracted and restructured into a MySQL database. The data chosen were the following:

- Class number (field 153, subfield a);
- Heading (field 153, subfield j);
- Relative index term (persons 700, corporates 710, meetings 711, uniform title 730, chronological 748, topical 750, geographic 751; with subfields);
- Notes for disambiguation: class elsewhere and see references (253 with subfields);
- Scope notes on usage for further disambiguation (680 with subfields); and,
- Notes to classes that are not related but mistakenly considered to be so (353 with subfields).

The total of 14,413 unique classes was extracted, of which 819 three-digit classes were found in the LIBRIS data collection described below.

3.2 Data Collection

The dataset of 143,838 catalogue records was derived from the Swedish National Union Catalogue, LIBRIS, which is the joint catalogue of the Swedish academic and research libraries. It was harvested using the Open Archives Initiative Protocol for Metadata Harvesting (OAIPMH)[2] in the period from 15 April to 21 April 2018.

[2] https://www.openarchives.org/OAI/openarchivesprotocol.html

LIBRIS makes its data available in the MARCXML format[3]. MARCXML is an XML Schema based on MARC (MAchine Readable Cataloguing) format for bibliographic data, derived from the ISO 2709 standard titled "Information and documentation -- Format for information exchange" used to exchange electronic records between libraries.

In total 143,838 records with unique id numbers, containing a DDC class (i.e. with MARC field 082[4]), were harvested. The records were parsed and all fields and subfields considered relevant were saved in an SQL-database, one field/subfield per row. Relevant fields were the following ones:

- Control number (MARC field 001), unique record identification number;
- Dewey Decimal Classification number (MARC field 082, subfield a);
- Title statement (MARC field 245, subfield a for main title and subfield b for subtitle); and,
- Keywords (a group of MARC fields starting with 6*), where available -- 85.8% of records had at least one keyword.

The records were formatted into an SQL table containing the total of 1,464,046 rows where each row contained 4 columns: ID, field, subfield, and value. The dataset had to be further pruned and cleaned before it could be used for classification experiments. All text features were stripped from special symbols, with the exception of the &-symbol which was replaced by the Swedish word for *and* (*och*), leaving only letters and numbers in the data. For each record, values for title, subtitle and keywords were concatenated into a list of words separated by whitespace, a process known as tokenization.

In the sample, only records containing DDC classes truncated to a three-digit code, ranging from 001 to 999, were used. Records with other codes as well as those missing both title and subtitle were excluded from the dataset. Duplicates (records with identical title + subtitle) were also removed. This cleaning phase resulted in a total of 143,838 records spread over 816 classes; or, 121,505 records spread over 802 classes when extracting only records which contained at least one keyword.

From the cleaned LIBRIS data a number of datasets were generated, which are presented in Table 1 below. One difficulty with the LIBRIS data is the extreme imbalance between DDC classes, the problem recognized also in previous research (see section 2 Background). The most frequent class is 839 (Other Germanic literatures) with 18,909 records, while 594 classes have less than 100 records (70 of those have only one single record). To see how this class imbalance affects classifiers, we have also generated a dataset containing only classes with at least 1,000 records, called major classes below. The latter resulted in 72,937 records spread over 29 classes, and 60,641 records spread over 29 classes when selecting records with keywords.

---

[3] https://www.loc.gov/standards/marcxml/

[4] For a list and description of MARC fields, see http://www.loc.gov/marc/bibliographic/.

Table 1. The different datasets generated from the raw LIBRIS data.

Dataset	ID	Records	Classes
Titles	T	143,838	816
Titles and keywords	T_KW	121,505	802
Keywords only	KW	121,505	802
Titles, major classes	T_MC	72,937	29
Titles and keywords, major classes	T_KW_MC	60,641	29
Keywords only, major classes	KW_MC	60,641	29

3.3 Machine Learning

Machine learning is the science of getting computers to learn, and improve their learning over time in autonomous fashion, by feeding them data. Instead of explicitly programming a computer what to do, the computer learns what to do by observing the data.

To automatically classify a resource, we need to build models that map input features, i.e. title, subtitle and, optionally, keywords, to a DDC class. These models learn from known, already classified, data (the LIBRIS database) and can later be used to automatically classify new resources. This is referred to as a supervised learning problem; both input features and correct classifications are known.

Machine learning algorithms cannot work with text data directly, so the list of words representing each record in the dataset needs to be encoded as a list of integer or floating point values (referred to as vectorization or feature extraction). The most intuitive way to do so is the 'bag of words' representation. The 'bag' contains all words that occur at least once in the dataset. A record in the dataset is represented as a vector with the number of occurrences for each word in the title, subtitle and, optionally, keywords. Since the number of distinct words is very high, the vector representing a record is typically very sparse (most values are 0). For the dataset with titles and subtitles, the bag contains a total of 130,666 unique words, and for the dataset with titles, subtitles and keywords, the bag comprises 134,790 unique words. Stopwords are not removed and stemming is not used.

When counting occurrences of each single word, all information about relationships between words in the data is lost. This is typically solved using n-grams. An n-gram is a sliding window of size n moving over a list of words, at a pace of one word forward in each step. If a 2-gram is applied, combinations of two words are used as input features instead of, or in combination with, single words (unigrams). For example the text

"machine learning algorithm" contains unigrams "machine", "learning", "algorithm", and 2-grams "machine learning" and "learning algorithm". Using n-grams drastically increases the size of the bag, but can possibly give better classification performance of models. Using unigrams and 2-grams for the datasets with titles, subtitles and keywords as input increases the size of the bag from 134,790 to 828,122 words/word combinations.

However, only counting occurrences is problematic: records with longer inputs (title, subtitle and, optionally, keywords) will have higher average count values than records with shorter inputs, even if they belong to the same DDC class. To get around this problem, the number of occurrences for each word is divided by the total number of words in the record, referred to as *term frequency (TF)*. A further improvement is to downscale weights for words that occur in many records and are therefore less informative than words that occur in only a few records. This is referred to as *inverse document frequency (IDF)*. Typically both of these approaches are used, called *TF-IDF conversion*.

The pre-processing of the text inputs results in high-dimensional, sparse input vectors of either integer values (counting occurrences only) or floating point values (TF-IDF conversion). Many machine learning algorithms are not suited for this type of input data, leaving only a few options left for our task. Historically, good results for different text classification tasks have been achieved with the Multinomial Naïve Bayes (NB) and Support Vector Machine with linear kernel (SVM) algorithms [22-24]. SVM typically gives better results than NB, but is slower to train. In addition, of the 143,838 records, 98.6% had one assigned DDC class and 1.4% had more than one assigned class. Because of this, the choice of machine learning algorithms was to apply those producing single output. Classifying records into multiple classes (multi-output classification) is a very different classification problem which severely limits the choice of possible machine learning algorithms. SVM would not be possible to use if we modelled the problem as a multi-output classification problem. Instead, multi-output classification will be approached in further research.

4 Results

Table 2 below shows classification accuracy (amount of records classified into the correct DDC class divided by the total number of records) of the NB classifier and Table 3 shows classification accuracy of the SVM classifier on the different datasets, using only unigrams or unigrams and 2-grams as inputs and TF-IDF conversion.

The columns labeled "Training set" show results when training and evaluating a classifier on all records in the dataset. This gives an indication of how effectively we can map inputs to classes, but does not show the generalization capabilities of the classifiers, i.e. how good they are at classifying records they have not seen before. Therefore, we have also trained the classifiers on 95% randomly selected records from the dataset, and used the remaining 5% of the records for evaluation (shown in the columns labeled "Test set").

Table 2. Accuracy of the Multinomial Naïve Bayes classifier on the different datasets.

Dataset	Accuracy, unigrams		Accuracy, unigrams + 2-grams	
	Training set	Test set	Training set	Test set
T	83.54%	34.89%	95.82%	34.15%
T_KW	90.01%	55.33%	98.14%	55.45%
KW	75.28%	59.15%	84.95%	58.11%
T_MC	90.83%	54.21%	98.63%	50.51%
T_KW_MC	95.42%	76.52%	99.66%	75.96%
KW_MC	86.94%	77.25%	94.24%	77.09%

Table 3. Accuracy of the Support Vector Machine classifier on the different datasets.

Dataset	Accuracy, unigrams		Accuracy, unigrams + 2-grams	
	Training set	Test set	Training set	Test set
T	93.74%	40.91%	99.59%	40.45%
T_KW	97.50%	65.25%	99.90%	66.13%
KW	83.09%	64.02%	92.38%	64.09%
T_MC	93.95%	57.99%	99.62%	57.80%
T_KW_MC	97.89%	80.75%	99.93%	81.37%
KW_MC	90.58%	79.56%	96.30%	80.38%

The best results were achieved when combining titles and keywords as input. Using only titles as input results in considerably worse accuracy than when combining titles and keywords or using only keywords as input, a difference around 22-23 percentage units for the major classes datasets. The results show that keywords have much higher information value than titles.

As expected, SVM has higher accuracy scores than NB on all datasets. This is in line with previous research on bibliographic data [23]. The best result for SVM when using all classes was 99.90% accuracy on the training set and 66.13% on the test set, when using both unigrams and 2-grams. When removing all classes with less than 1,000 records, the accuracy on the test set increased to 81.37%.

The overall lower accuracy scores on the test sets compared to the training set even when removing classes with few records may be affected by a phenomenon called indexing consistency. A number of studies have shown that humans assigning classes or keywords to bibliographic records often do this in an inconsistent manner, both compared to themselves (intra-indexing consistency) and compared to other humans (inter-indexing consistency) [25]. Since the classifiers learn from LIBRIS data categorized by humans, this inconsistency may affect their generalization capabilities leading to difficulties when classifying records they have not seen before. The extreme class imbalance also affects the generalization capabilities negatively.

Combining both unigrams and 2-grams only marginally improved the results on the test sets. The highest accuracy was achieved when using SVM and both titles and keywords as input and only major classes. For this dataset the accuracy only increased with 0.62 percentage units when combining unigrams and 2-grams. For NB, the accuracy scores were for most datasets lower than when using unigrams only.

To summarize, using keywords or combining titles and keywords gives much better results than using only titles as input. SVM outperforms NB on all datasets, and the class imbalance where many DDC classes only have few records greatly affects classification performance. Combining unigrams and 2-grams in the input data only marginally improved classification accuracy but leads to much longer training times.

4.1 Stemming

Stemming is the process of reducing words to their base or root form. There are several stemming algorithms that can be used, for example lemmatisation or rule-based suffix-stripping algorithms. To investigate how stemming affects accuracy, we have generated two new datasets where the Snowball stemming algorithm for Swedish was used on titles and subtitles[5]. No stemming was used on keywords as they are typically already in base form. We confirmed that this was a good choice by running some tests which showed that accuracy decreased when using stemming on keywords.

Table 4 shows classification accuracy of the Naïve Bayes classifier and Table 5 shows classification accuracy of the Support Vector Machine classifier on the datasets using stemming on titles and subtitles. The results show that stemming only has minor effect on classification accuracy, likely because titles have low information value. The accuracy of the best model using major classes only (SVM using titles and keywords as input, and combining unigrams and 2-grams) increased from 81.37% without stemming to 81.80% when stemming was used, an increase with 0.43 percentage units.

[5] http://snowball.tartarus.org/algorithms/swedish/stemmer.html

When using only titles as input, the accuracy gain was slightly smaller, 0.30 percentage units. When all classes are used, the accuracy gain is even lower, 0.16 percentage units for SVM using titles and keywords as input.

Table 4. Accuracy of the Multinomial Naïve Bayes classifier when using stemming.

Dataset	Accuracy, unigrams		Accuracy, unigrams + 2-grams	
	Training set	Test set	Training set	Test set
T_stm	78.07%	35.09%	94.36%	34.62%
T_KW_stm	88.68%	55.76%	97.95%	56.44%
T_MC_stm	86.44%	55.55%	97.68%	52.56%
T_KW_MC_stm	94.32%	76.36%	99.59%	76.36%

Table 5. Accuracy of the Support Vector Machine classifier when using stemming.

Dataset	Accuracy, unigrams		Accuracy, unigrams + 2-grams	
	Training set	Test set	Training set	Test set
T_stm	89.15%	41.17%	98.88%	40.96%
T_KW_stm	96.71%	65.16%	99.87%	66.29%
T_MC_stm	89.94%	57.80%	98.99%	58.10%
T_KW_MC_stm	97.21%	81.07%	99.91%	81.80%

5 Concluding Remarks

Achieving high accuracy of 81% reported when using SVM has proven to be dependent on the availability of a good amount of training data, i.e. at least 1,000 records per class. The lack of training data for a large number of classes is even more severe when looking at more specific classes beyond the top three levels; here out of 14,413 available DDC classes, only about 6% were possible to take into consideration.

Therefore, an approach which could be tested in the future is to combine SVM with a string matching algorithm, relying on a large number of synonyms and related terms which exist in the DDC, as well as making use of hierarchies for disambiguation. This is in line with previous research that has demonstrated good results using this approach, albeit in different contexts. Another direction for future work could be to try hierarchical classification, where one classifier is trained on the first 10 top-level categories, and individual classifiers are trained on data only belonging to one top-level category.

More generally, further research is needed to gain a scientifically sound understanding of the level to which it is possible to apply automatic classification with the DDC to Swedish resources. Because of factors such as low indexing consistency existing metadata records cannot be used as 'the gold standard': the classes assigned by algorithms (but not human-assigned) might be wrong or might be correct but omitted during human indexing by mistake. The proposed future research involves a more comprehensive approach to 'gold standard' production [11].

Acknowledgements

Thanks are due to OCLC which provided the project with electronic DDC, Swedish version. We are very grateful to Rebecca Green and Sandi Jones for all their advice on how to best process and use the electronic DDC files.

References

1. Lancaster, F. W. Indexing and Abstracting in Theory and Practice. Facet: London, 2003.
2. Hunter, R. N. "Successes and Failures of Patrons Searching the Online Catalog at a Large Academic Library: A Transaction Log Analysis." RQ, 30(3), 395-402, 1991.
3. Villen-Rueda, L., J. A. Senso and F. De Moya-Anegon. "The Use of OPAC in a Large Academic Library: A Transactional Log Analysis Study of Subject Searching." The Journal of Academic Librarianship,33(3), 327-337, 2007.
4. Meadow, K., and J. Meadow. "Search Query Quality and Web-Scale Discovery: A Qualitative and Quantitative Analysis." College & Undergraduate Libraries 19(2-4), 163-75, 2012.
5. Svanberg, M. Dewey Projektet: slutrapport. 2013. Available at: http://www.kb.se/Dokument/Deweyprojektet%20slutrapport%2020130614.pdf
6. Roitblat, H. L., A. Kershaw, A., and Oot, P. "Document Categorization in Legal Electronic Discovery: Computer Classification vs. Manual Review." Journal of the American Society for Information Science and Technology 61(1), 70-80, 2010.
7. Anderson J. and J. Perez-Carballo. "The Nature of Indexing: How Humans and Machines Analyze Messages and Texts for Retrieval: Part II: Machine Indexing, and the Allocation of Human Versus Machine Effort." Information Processing and Management 37, 255-277, 2001.
8. Svarre, T. J. and M. Lykke. "Simulated Work Tasks: The Case of Professional Users." In Proc. 5th Information Interaction in Context Symposium, 215-218, 2014.

9. Silvester, J. P. "Computer Supported Indexing: A History and Evaluation of NASA's MAI System." In Encyclopedia of Library and Information Services 61(24), 76-90, 1997.
10. Ruiz, M. E., A. R. Aronson, and M. Hlava. "Adoption and Evaluation Issues of Automatic and Computer Aided Indexing Systems." In Proceedings of the American Society for Information Science and Technology, 45(1), 1-4, 2008.
11. Golub, K. et al. "A Framework for Evaluating Automatic Indexing or Classification in the Context of Retrieval." Journal of the Association for Information Science and Technology (JASIST), 67(1), 3-16, 2016.
12. Golub, K. "Automated Subject Classification of Textual Web Documents." Journal of Documentation 62(3), 350-371, 2006.
13. Golub, K. Automatic subject indexing of text. ISKO Encyclopedia of Knowledge Organization, 2017. Available at: http://www.isko.org/cyclo/automatic.
14. Svenonius, E. The Intellectual Foundation of Information Organization. Cambridge, Mass.: MIT Press, 2000.
15. Chen, H. and S. Dumais. "Bringing Order to the Web: Automatically Categorizing Search Results." In Proceedings of the ACM International Conference on Human Factors in Computing Systems, Den Haag, 145-152, 2000.
16. Wang, J. "An Extensive Study on Automated Dewey Decimal Classification." Journal of the American Society for Information Science and Technology, 60(11), 2269-2286, 2009.
17. Lösch, M., U. Waltinger, W. Hortsmann and A. Mehler, A. "Building a DDC-annotated Corpus from OAI Metadata." Journal of Digital Information, 12(2), 2011.
18. Thompson, R., K. Shafer and D. Vizine-Goetz. "Evaluating Dewey Concepts as a Knowledge Base for Automatic Subject Assignment." In Proc. Second ACM Int. Conf. on Digital libraries (DL '97), 37-46, 1997.
19. Khoo, M. et al. "Augmenting Dublin Core Digital Library Metadata with Dewey Decimal Classification." Journal of Documentation, 71(5), 976-998, 2015.
20. Golub, K. "Automated Subject Classification of Textual Documents in the Context of Web-based Hierarchical Browsing: PhD thesis." Lund: Department of Electrical and Information Technology, Lund University, 2007.
21. Golub, K., T. Hamon and A. Ardö. "Automated Classification of Textual Documents Based on a Controlled Vocabulary in Engineering. Knowledge Organization, 34(4), 247-263, 2007.
22. Wang, J. "An Extensive Study on Automated Dewey Decimal Classification". Journal of the American Society for Information Science and Technology, 60, 2269-2286, 2009.
23. Trivedi, M., S. Sharma, N. Soni and S. Nair. "Comparison of Text Classification Algorithms". International Journal of Engineering Research & Technology, 4(2), 2015.
24. Aliwy, A. H. and E.H. Abdul Ameer. "Comparative Study of Five Text Classification Algorithms with their Improvements". International Journal of Applied Engineering Research, 12(14), 2017.
25. Leininger, K. "Interindexer Consistency in PsycINFO". Journal of Librarianship and Information Science, 32(1), 4-8, 2000.