Enhancing Document Modeling by Means of Open Topic Models
Crossing the Frontier of Classification Schemes in Digital Libraries by Example of the DDC

Alexander Mehler and Ulli Waltinger

Text Technology
Bielefeld University
Universitätsstraße 15, 33602 Bielefeld, Germany
Alexander.Mehler@uni-bielefeld.de, Ulli.Waltinger@uni-bielefeld.de

Abstract. We present a topic classification model using the Dewey Decimal Classification (DDC) as the target scheme. This is done by exploring metadata as provided by the Open Archives Initiative (OAI) to derive document snippets as minimal document representations. The reason is to reduce the effort of document processing in digital libraries. Further, we perform feature selection and extension by means of social ontologies and related web-based lexical resources. This is done to provide reliable topic-related classifications while circumventing the problem of data sparseness. Finally, we evaluate our model by means of two language-specific corpora. This evaluation shows that DDC-related classifiers come into reach which mainly explore OAI metadata.

Key words: closed topic models, open topic models, document modeling, document snippets, DDC, OAI.

1 Introduction

It is beyond any doubt that automatic content classification is of outmost interest in digital libraries [15]. The idea is to provide content-related add-ons which allow for improving retrieval and document processing. In this introduction, we give a short overview of competing approaches in this field of research which focus on condensed document representations as provided, for example, by keyword lists or summaries.

An early approach to clustering document summaries at different levels of thematic granularity is the scatter-gather method [5, 10]. In recent years, variants of the Suffix Tree Clustering (STC) algorithm [21, 36, 44, 35] also attracted attention in this field of research. These variants explore common sub-phrases of documents which are judged to be similar because of their common suffix trees. An alternative approach with a focus on hierarchical document classification has been introduced by [45] who explores search query snippets instead of summaries as the main source of document representation. These and related
approaches form the core of search engines as, e.g., *Vivísimo* [38], *Mapuccino* [16] and *Carrot* [25], which perform post-retrieval document clustering. That is, they detect topic labels of thematic clusters based on document snippets (e.g., titles) as retrieved by search queries [13]. The idea behind this approach is to enhance the identification of relevant documents by eliminating the need to skim large numbers of irrelevant texts.

This approach is easily transferred to the area of digital libraries where document snippets are given by subject-related metadata. A metadata protocol which recently became more and more prominent is the *Open Archives Initiative-Protocol for Metadata Harvesting* (OAI-PMH). This protocol implements a standardized metadata model for facilitating exchange between repositories. Approaches to document clustering in digital libraries have focused, among other things, on extending search queries and metadata entries of documents [8, 28]. In this case, clustering is performed to detect the subject area of documents based on a predefined classification scheme, that is, a closed topic model [23].

In this article, we present a topic classification model which uses the *Dewey Decimal Classification* (DDC) [24] as the target scheme. Our approach is novel in two senses. On the one hand, we use metadata as provided by the *Open Archives Initiative* (OAI) to derive document snippets as minimized document representations. This is done to reduce the time and space complexity of document processing. On the other hand, we perform feature selection and feature extension by means of social ontologies and related web-based lexical resources. This is done to provide reliable topic-related classifications while circumventing the problem of data sparseness. In a nutshell, the article provides a model of topic-related document classifications whose semantics is explored by means of web-based resources of semantic relatedness and whose document model is mainly based on OAI data.

The article is structured as follows: in Section 2, we describe several reference points of document modeling in digital libraries. We do that to shed light on how to cross the frontier of classification schemes, i.e., moving from closed topic models toward open topic models. Next, in Section 3, we describe our test corpora and the representation of documents by means of OAI metadata. In Section 4, we introduce a search engine-based classifier for the DDC which integrates social semantic knowledge to enhance document representation. Further, in Section 5, we present an experiment in DDC classification using two different corpora and five different DDC-related classifiers. This experiment is discussed in detail in Section 6. Finally, Section 7 concludes and suggests prospects for future work.

2 On Reference Points of Document Modeling

When classifying a document by its topic, one has at least two possibilities: either one uses a closed, i.e., fixed system of categories (e.g., a classification scheme) or one uses an open system that changes in time. As all systems change in the long run, we have to avoid a triviality by stipulating that the time scale for open systems is less than the rate of change of classification schemes. In order to arrive
at a working definition, we assume that this scale manifests a random division of time as given by the change rate of social ontologies [17, 22]. From this point of view, there are two reference points of system dynamics of which only the second justifies the attribute open:

- Firstly, because a system of topic categories may be given by a category graph or, more narrowly, by a tree (as, e.g., the DDC), its change may affect only its links leaving its vertex set untouched.
- Secondly, the vertex set of the graph may change in time by the deletion, merging, splitting or insertion of categories. In this case, a change of the set of links may, but does not need to be a consequence of the system dynamics.

Note that if we do not deal with graph-like topic systems, but with category sets (as done by the majority of approaches to text categorization [33]), the latter distinction is irrelevant.

In any event, the distinction of open and closed models is related to a second, more classical difference: that is, documents may be similar by topic or by genre. Take the example of nanotechnology dealt with by a journal article in contrast to a doctoral thesis. In this case, the documents agree on topic but differ on genre. Conversely, we can have documents of the same genre, which differ on topic. Thus, we get two, not necessarily orthogonal views of document classification [1, 9] and, hence, a decision matrix by which four document models can be distinguished (cf. Table 1):

- Closed Topic Model (CTM): As mentioned above, a topic model is closed if its composition is fixed. A CTM is given by a classification scheme (e.g., the DDC or MeSH) as a terminological ontology [34] whose vertices denote
conceptual types of topic areas. Such schemes are generated by a small number of selected experts whose collaboration is controlled according to the prospected target ontology communicated to its users in a one-to-many setting. The low rate of change of CTMs corresponds to a non-random time scale. This is somehow in contradiction to the dynamics and openness of the human topic universe with its ever emergent and growing topics. However, a CTM guarantees repeatability of classification results and comparability over time so that it is still usable in digital libraries. Note that closed topic models are suitable as target models of supervised learning as their fixed nature is a precondition of persistent and reliable training data.

– **Open Topic Model (OTM):** In an OTM, the topic categories are not enumerated in advance (as in supervised learning) or the result of labeling clusters found in a fixed set of empirical data (as in unsupervised learning). Rather, OTMs explore topic labels from an open, that is, ever-growing social ontology. Social ontologies as, e.g., the category system of Wikipedia, are output by social tagging [22]. They emerge as a solution to a coordination problem among large groups of interacting agents [2]. This relates to the sharing of a collectively structured semantic universe in the form of non-formal ontologies [17]. Unlike the one-to-many communication of terminological ontologies, social ontologies result from a many-to-many communication in which groups of agents interact to constitute and organize a dynamically growing universe of content units. Social ontologies provide large-scale and flexible knowledge systems for building OTMs, and these evolve according to the time scale of the topic universes of speech communities. In a nutshell: an OTM obtains its topic model from a social ontology with which it co-evolves. OTMs extend the paradigm of supervised and unsupervised learning by integrating human computation. In line with this model are web-based resources of lexical relatedness as explored by *collocation networks* [11]. By analogy to OTMs, collocation networks grow due to the dynamics of human computation in the web and also dispense with predefining any semantics. This holds all the more for measures of semantic relatedness based on search engines which directly access the web as a, so to speak, universal information base. In this article, we explore these three different resources of OTMs in a single framework.

– **Closed Genre Model (CGM):** topic models are predominant in IR. However, there is rising interest in alternative retrieval models, e.g., by taking genre into account [7, 20, 29]. This is mostly done by CGMs whose categories are enumerated in advance [37]. Although a standardization of genre categories by analogy to the stringency of CTMs is far away, the web mining community establishes such systems by so-called *genre palettes* [32] to guarantee comparability of classification results [27].

– **Open Genre Model (OGM):** From that perspective, one may think of genre palettes which co-evolve with some social tagging. At first glance, this seems to be a futile endeavor as textual genres change much more slowly than topics. However, if we think, e.g., of games with the purpose of annotating
Table 2. The 10 top categories of the DDC (left) and the 10 subdivisions of the class 000 (right) [24].

<table>
<thead>
<tr>
<th>No</th>
<th>Label</th>
<th>No</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Computer science, information &amp; general works</td>
<td>000</td>
<td>Computer science, knowledge &amp; systems</td>
</tr>
<tr>
<td>100</td>
<td>Philosophy &amp; psychology</td>
<td>010</td>
<td>Bibliographies</td>
</tr>
<tr>
<td>200</td>
<td>Religion</td>
<td>020</td>
<td>Library &amp; information sciences</td>
</tr>
<tr>
<td>300</td>
<td>Social sciences</td>
<td>030</td>
<td>Encyclopedias &amp; books of facts</td>
</tr>
<tr>
<td>400</td>
<td>Language</td>
<td>040</td>
<td>[Unassigned]</td>
</tr>
<tr>
<td>500</td>
<td>Science</td>
<td>050</td>
<td>Magazines, journals &amp; serials</td>
</tr>
<tr>
<td>600</td>
<td>Technology</td>
<td>060</td>
<td>Associations, organizations &amp; museums</td>
</tr>
<tr>
<td>700</td>
<td>Arts &amp; recreation</td>
<td>070</td>
<td>News media, journalism &amp; publishing</td>
</tr>
<tr>
<td>800</td>
<td>Literature</td>
<td>080</td>
<td>Quotations</td>
</tr>
<tr>
<td>900</td>
<td>History &amp; geography</td>
<td>090</td>
<td>Manuscripts &amp; rare books</td>
</tr>
</tbody>
</table>

multimedia objects [40], we enter the required dynamics: if we apply this model to the area of emergent web genres, we arrive at a scenario by analogy to OTMs. In this sense, OGMs emerge as a way forward that addresses the deficient coverage rate of genre palettes with respect to the dynamics of web-based communication.

So far we have related topic and genre to either open or closed category systems. To complement this picture, we distinguish, by analogy to [9], two levels of realization of topic and genre: the lexical micro structure of a document and its (e.g., rhetorical, functional or logical) macro structure (cf. Figure 1). The vast majority of approaches to IR explore easily accessible lexical structures. Only a minority utilize document macro structures [cf. 6, 18].

In this article we introduce a document representation model which combines a closed with an open topic model. This is done by exploring the lexical structure of a document subject to a highly restricted representation of its macro structure. As a closed topic model, we utilize the DDC in combination with, among other things, the Wikipedia as the operative social ontology. More specifically, we explore the OAI metadata of a document as a highly condensed document representation where the Wikipedia and web-based lexical resources are used to circumvent the problem of data sparseness and to secure the usage of topic-related, that is, semantic document features. In a nutshell, this article crosses the border of closed topic models into the direction of open topic models to profit from both the openness and covering rate of the latter and the systematicity of the former. This is what we subsume under the notion of social semantics for digital libraries.

3 Building a Test Corpus for Closed Topic Models

In this section, we describe the CTM used for document classification, i.e., the Dewey Decimal Classification (DDC) (cf. Section 3.1). Further, we describe the preprocessing of input documents by their OAI metadata (cf. Sec. 3.2 and 3.3).

---

1 See www.websitewiki.de for an example of social software used to describe websites.
3.1 The DDC as a Closed Topic Model

As a CTM we use the DDC [24] which is the most common classification scheme for subject cataloguing in libraries. The DDC starts from 10 main classes which are subdivided on two levels such that each subdivision is structured into 10 areas (cf. Table 2). As a result, a forest of 10 trees is spanned each of which has 111 vertices — a remarkably artificial ontology as is typical for CTMs. We access the DDC by means of the Bielefeld Academic Search Engine (BASE) [26] which provides OAI metadata of input documents. Among other things, this includes their DDC-related classifications. BASE has mapped documents on all levels of the DDC, with up to 100,000 training candidates processed so far. These classifications allow us to build a training corpus for the top-level categories of the DDC, and these, in turn, are used finally to evaluate our approach.

3.2 Minimizing Representation Effort by Exploring OAI Snippets

A central demand of digital libraries concerns the fast, online and reliable classification of documents. In order to guarantee reliability, the documents need to be processed carefully. This requirement is opposed by the space complexity of processing documents up to the length of theses. On the other hand, if one reduces the data to be processed, the problem of data sparseness is raised which may produce misclassifications. In order to balance the prevention of the latter risk against the former requirement, we do not process input documents directly, but explore their Open Archives Initiative (OAI) metadata (cf. Table 3 for a sample document representation). More specifically, we use the OAI-Protocol for Metadata Harvesting (OAI-PMH) to access document snippets in the form of OAI metadata fields. This allows us to explore document data without the need to parse the entire document.

Generally speaking, OAI represents a document by a title, creator, subject, description, publisher, type, format, and language tag (cf. Table 3).
Table 4. Corpus statistics: standard deviation and arithmetic mean of the description and subjects fields of OAI-snippets in the test corpus. Counting is based on tokens as the counting units.

<table>
<thead>
<tr>
<th>Info</th>
<th>English</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description: arithmetic mean</td>
<td>84.45</td>
<td>15.43</td>
</tr>
<tr>
<td>Description: standard deviation</td>
<td>9.36</td>
<td>10.81</td>
</tr>
<tr>
<td>Subjects: arithmetic mean</td>
<td>7.74</td>
<td>12.75</td>
</tr>
<tr>
<td>Subjects: standard deviation</td>
<td>7.03</td>
<td>10.51</td>
</tr>
</tbody>
</table>

These tags are mainly based on the Dublin Core metadata element set extended by a small set of OAI-specific tags. For the task of document representation, we further reduce this set so that each document is finally represented by three (types of) tags: that is, **title**, **subject**, and **description**. From that perspective, our classification hypothesis reads as follows:

**H1** The topic of a scientific document is reliably classified by processing its title, subject fields and short description.

In relation to the DDC as the operative CTM we get the following target statement to be evaluated experimentally: *you shall know the top-level DDC class of a scientific document by its OAI snippets*. Note that the description field of the English documents included into our experimental corpus contains 84.45 tokens on average, while its subject field contains on average 7.74 tokens (cf. Table 5) — *this is a remarkably small set of tokens in relation to documents of the length of articles or even books*. Note also that apart from the title, none of the OAI fields taken into consideration is necessarily extracted from the underlying document. Rather, these fields may depart in their lexical structure from the lexical structure of the document itself.

Our minimization procedure reduces the space complexity of document representation and saves processing time. However, it also raises the risk of data sparseness. To face this risk, that is, to secure semantic reliability of the features which are finally explored to classify documents, we cross the border to open topic models and utilize a social ontology in conjunction with web-based resources of lexical relatedness. This is described *in extenso* in Section 4. First, however, we describe the preprocessing of the document snippets.

### 3.3 Preprocessing OAI Snippets

The preprocessing of OAI document snippets is performed in the usual way. That is, we perform language identification, segmentation of the logical document structure (including sentence boundary detection), lemmatization of lexical units, part of speech tagging and named entity recognition [19, 41]. This allows for filtering out non-lexical tokens as well as function words. As a result, we
get a linguistically tagged input stream of lexical features per document snippet which is input to the next step: feature selection. This step is performed by mapping an input stream of lexical units of the OAI snippet representation \( x \) of a document \( X \) onto a fuzzy set \( \mathcal{X} \) where, for any lexical item \( a \) in \( x \), the membership value \( \mu_{\mathcal{X}}(a) \) is computed as the frequency of \( a \) in \( x \) standardized by the frequency of the most frequent item in \( x \). This allows us to build a ranked feature list per input document \( X \), where the higher \( \mu_{\mathcal{X}}(a) \), the higher the rank of feature \( a \). From a technological point of view, \( \mathcal{X} \) is an algebraic representation of a sparse matrix as represented, e.g., in SVMlight [12]. Thus, we easily derive a vector representation of \( \mathcal{X} \), weight it in the usual way [31], derive a Vector Space Model (VSM) [30] of the input corpus and finally make this an input to Latent Semantic Analysis (LSA) [14] which serves as a baseline scenario in our experiment (cf. Section 5).

4 Two Novel DDC Classifiers

As a novel method of classifying OAI-Metadata according to the DDC we now introduce a generalised Search Engine Quotient (SEQ). This classifier extends the so-called Google quotient [3, 4] by combining distance measuring with a category feature model based on co-occurrence statistics. Since we do not focus on measuring the relatedness of pairs of tokens but on classifying documents by means of OAI snippets, we need to explore for features in these snippets (Section 4.1) as significant content descriptors (Section 4.2). Further, we need to define a separate feature model for each of the categories to be classified in order to relate them to our document feature models (Section 4.3). Next, we have to implement a search engine-based quotient — in the present case by means of the Wikipedia and the search engine Yahoo — in order to map OAI snippets to DDC classes (Section 4.4). By this procedure we get a classification value for each main class of the DDC which expresses the relatedness of a given OAI-input stream to the selected class.

4.1 Building Document Models

Generally speaking, we view the OAI data assigned to a document as a highly condensed representation of that document. More specifically, apart from function words we view any lexical constituent \( a_i \) of the snippet \( S_j \) assigned to a document \( D_j \) as a candidate feature of the content of that document. By ranking these lexical constituents according to their standardized term-frequency (tf) in descending order we get information about the most significant content terms of \( D_j \). That is, terms \( a_i \) are ranked with respect to documents \( D_j \) by computing their frequency index \( tf_{ij} \) where \( f_{ij} \) is the frequency of \( a_i \) in \( D_j \) and \( L(D_j) \) is the set of all lexical constituents of \( D_j \) (note that function words are excluded):

\[
    tf_{ij} = \frac{f_{ij}}{\max_{a_k \in L(D_j)} f_{kj}} \in (0, 1]
\]  

(1)
Additionally, we take multi-word units and phrases into account by ranking them according to their standardized phrase-frequency ($pf$). Note that we explore frequent phrases by means of $n$-grams of tokens. This allows us to rank phrases $p_i$ with respect to documents $D_j$ by computing their frequency index $pf_{ij}$

$$pf_{ij} = \frac{f_{ij}}{\max_{p_k \in P(D_j)} f_{kj}} \in (0, 1]$$

where $f_{ij}$ is the frequency of $p_i$ in $D_j$ and $P(D_j)$ is the set of all $n$-grams of $D_j$. By the rank-frequency lists of words and phrases assigned to a document $D_j$ we can select the topmost ranked features of both lists. This is done by means of two stacks: the word stack $WS_j$ and the phrase stack $PS_j$ which list all words and phrases of $D_j$ in descending order of their standardized frequencies $tf_{ij}$ and $pf_{ij}$, respectively. In a nutshell: $WS_j$ ($PS_j$) is the list of all lexical (phrasal) items of $D_j$ in descending order of their significance as content descriptors of $D_j$ where significance is measured in terms of frequency.

### 4.2 Feature Verification

By means of the feature stacks $WS_j$ and $PS_j$ assigned to a document $D_j$ we can select the topmost ranked lexical and phrasal features of $D_j$. This is done in order to secure reliable search results when using document features as search terms of a search engine-based query. The number $N$ of features to be selected in this step has to be carefully chosen. By selecting too few features, search results get unspecific, while too many features can distract the search from the actual content of $D_j$. In order to reduce the risk of choosing the wrong number $N$ of selected features we utilize and refine the approach of [43]. More specifically, we initially set $N = \max(|WS_j|, |PS_j|)$ where $|S|$ is the length of stack $S$. Then, we perform a search in our reference search engine — in the present case Wikipedia. This is initially done by using all $N$ topmost ranked lexical and phrasal features. Next, we decrement $N$, that is, $N \leftarrow N - 1$, and repeat the latter search till we get at least one Wikipedia article as a search result. That way, we select the $N \leq |WS_j|$ topmost ranked lexical and the $N \leq |PS_j|$ topmost ranked phrasal features without the need to preset $N$. In other words: we align the number $N$ of significant features to the characteristics of the given input document $D_j$. We denote this threshold by $N_j$. As a result, each document $D_j$ is represented by its $N_j$ most significant lexical and $N_j$ most significant phrasal features extracted from its OAI metadata representation. The reason to do this is to filter out irrelevant features even if they are frequent. The final feature set is denoted by

$$F(D_j) = \{F \in WS_j \mid \text{rank}(F) \leq N_j\} \cup \{F \in PS_j \mid \text{rank}(F) \leq N_j\}$$

which is the feature set representation of document $D_j$ where rank$(F)$ is the rank of feature $F$ in the corresponding stack as determined by the frequency index of $F$ — we write $F \in WS_j$ to denote that feature $F$ is on stack $WS_j$. 


4.3 Building Topic Models

So far we have shown how to represent documents by subsets of lexical and phrasal constituents. Now, we turn to the task of learning separate feature models for each of the 10 main classes of the DDC. More specifically, we represent each main class of the DDC by means of two resources of feature extraction: (i) the titles of the divisions and sections dominated by the corresponding class and (ii) web-based co-occurrence data related to these titles.

As an example, consider the first class of the DDC: 000 Computer Science, Information & General Works. This class dominates 10 divisions on the second level of the DDC (i.e., Bibliographies, Library, Encyclopedia, ...) and 100 sections on the third level (i.e., Knowledge, The book, Systems, Data processing, ...). Each of these division and section titles is added to the representation model of the class 000. That is, each of the 10 DDC classes is represented by 110 lexical items (including multiword terms). In the second step, we enrich this feature model by extending each feature by its most significant co-occurrence neighbor. In our experiments we retrieved this neighborhood information by the Web service of the Leipziger Wortschatz² [11]. In the case of class 000 we enriched, for example, the feature book by published and the feature system by operating as these are the most significant lexical neighbors of both features in the latter co-occurrence network. This approach overcomes problems of data sparseness by exploring co-occurrence data as it relies on two related feature resources: taxonomical information provided by the DDC and word association information provided by the co-occurrence network. Note that we add only one feature per DDC division and section title so that each main class is represented by 220 content descriptors. As a result, we get a feature set $F(C_i)$ of 220 descriptors per DDC main class $C_i$.

The next step is to compute for each class $C_i$ and each document $D_j$ an index of overlap of their representation models $F(C_i)$ and $F(D_j)$. As we deal with linguistic features we do that by accounting for composite units. More specifically, two (lexical or phrasal) features $F, G$ are said to overlap, that is, $F \sim G$, if either $F = G$ or if $F$ is a substring of $G$. Then, as an index of overlap of $F(C_i)$ with $F(D_j)$ we compute

$$\text{overl}(F(C_i), F(D_j)) = |\{F \mid \exists G \in F(C_i) \exists H \in F(D_j) F \sim G \land F \sim H\}| \tag{4}$$

Equation 4 tells us the degree to which features of a given input document $D_j$ occur in the feature representation of class $C_i$. This is the starting point of performing the final classification as explained in the next section.

4.4 The Classification Rule

The overlap index in Equation 4 relates only those classes and documents whose feature sets actually overlap. As a matter of fact, such an overlap is a strong indicator of class membership but occurs rather infrequently. Therefore, we need

² http://corpora.informatik.uni-leipzig.de/
a fall-back strategy which covers all cases in which this overlap does not occur or is not large enough to be indicative of class membership. As such, we utilize a search engine-based quotient which is computed as follows: for DDC class $C_i$ and any of its features $F_k \in \mathbf{F}(C_i)$ we compute the search engine-based relatedness $\text{rel}(F_k, D_j)$ of $F_k$ and document $D_j$ by

$$\text{rel}(F_k, D_j) = 2 \times \frac{g(\mathbf{F}(D_j), F_k)}{g(\mathbf{F}(D_j)) + g(F_k)}$$

(5)

where $g(\mathbf{F}(D_j), F_k)$ is the number of Yahoo hits one gets when using all features in $\mathbf{F}(D_j)$ together with $F_k$ as search terms while $g(\mathbf{F}(D_j))$ and $g(F_k)$ are the corresponding numbers of hits one gets when searching by the features in $\mathbf{F}(D_j)$ and by $F_k$ separately. Next, we sum up these values for different features $F_k \in \mathbf{F}(C_i)$ to relate $C_i$ and $D_j$ as a whole:

$$\text{Rel}(C_i, D_j) = \sum_{F_k \in \mathbf{F}(C_i)} \text{rel}(F_k, D_j)$$

(6)

Because it requires too much search effort to sum over all 220 features of class $C_i$ we only consider the 10 division titles assigned to $C_i$ — note that all other features are accounted for by the overlap index in Equation 4. Thus, for the 10 main classes and 10 divisions per class we perform $10 \times 10 \times 3 = 300$ search queries in the course of classifying a given document $D_j$ (note that $g(\mathbf{F}(D_j), F_k)$, $g(\mathbf{F}(D_j))$ and $g(F_k)$ are computed separately). Next, we compute an overall classification value which takes the index of search-engine-based relatedness of document $D_j$ and class $C_i$ into account as well as their overlap as computed by Equation 4:

$$\text{SEQ}(C_i, D_j) = \alpha \cdot \text{overl}(\mathbf{F}(C_i), \mathbf{F}(D_j)) + (1 - \alpha) \cdot \text{Rel}(C_i, D_j)$$

(7)

This index explores four resources: (i) the Wikipedia as a source of document feature selection, (ii) the DDC hierarchy as a source of category feature extraction, (iii) a web-based co-occurrence network for feature enrichment, and (iv) a search engine to provide a fall-back strategy. Note that $\alpha$ allows us to balance these different resources of computing the class membership of a document. However, in order to reduce the parameter set of our study we set $\alpha = .5$. Finally, we classify document $D_j$ by the class $C_i$ for which

$$C_i = \arg \max_{C_k \in \mathbf{C}} \{\text{SEQ}(C_k, D_j)\}$$

(8)

where $\mathbf{C} = \{C_1, \ldots, C_{10}\}$ is the set of the 10 main classes of the DDC.

4.5 Utilizing a Wikipedia-based OTM to Build a DDC-Related Classifier

To get a second DDC-related classifier, we explore the Wikipedia as a social-ontological resource of lexical features for modeling documents and topics. In
contrast to the SEQ-based classifier, the Wikipedia-based classifier omits the feature verification step. That is, it uses all lexical features to compute the membership value of a document to a DDC category. In this context, a reduced vector representation of the Wikipedia data set is used to measure the semantic relatedness of a lexical feature $F$ of the OAI snippet of a document $D$ to the corresponding DDC category $C$. We define the relatedness score $WR_X(F, C)$ of feature $F$ with respect to category $C$ as follows:

$$WR_X(F, C) = 1 - \left( \frac{\max\{\log(f_X(F)), \log(f_X(C))\} - \log(f_X(F \land C))}{\log M - \min\{\log(f_X(F)), \log(f_X(C))\}} \right)$$

(9)

where $f_X(x)$ is the document frequency, that is, the number of documents of the Wikipedia document collection $X$ in which $x$ occurs, and $M$ is the cardinality of this document collection. This score is either based on exploring the article graph — in this case $X = art$ — or on the category graph of the Wikipedia — i.e., $X = cat$. By balancing both sources of relatedness, that is, $WR_{art}$ and $WR_{cat}$, with the help of a parameter $\beta \in [0, 1]$, we get the following formula as an overall measure of the relatedness of $F$ and $C$:

$$WR(F, C) = \beta \cdot WR_{art}(F, C) + (1 - \beta) \cdot WR_{cat}(F, C)$$

(10)

This allows us to finally compute the relatedness of a document $D$ and a DDC category $C$ by a mean value:

$$WR(D, C) = |L(D)|^{-1} \sum_{F \in L(D)} WR(F, C)$$

(11)

where $L(D)$ is the set of OAI-based features of document $D$. Finally, we derive a classification rule by analogy to Expression 8:

$$C_i = \arg\max_{C_k \in C} \{WR(C_k, D_j)\}$$

(12)

See [42] for a thorough description of this approach.

5 Experimentation

In this section, we evaluate the classifiers of Section 4 in relation to baseline scenarios. This is done by classifying documents with respect to the top-level DDC categories based on OAI metadata representations. In this comparative study, we put special emphasis on the SEQ- and the Wikipedia-based classifier as approaches to crossing the frontier of classification schemes into the direction of OTMs. Further, we subdivide this experiment in two parts (cf. Table 5). In the first part we focus on English documents, while in the second part, we deal with German documents. As we will see in Section 6, the outcomes for both parts are quite different. However, this does not reflect a linguistic divergence, but is caused by a difference in the quality of the OAI metadata of the corresponding input documents.
For each language-specific part of our experiment, we evaluate five different classification algorithms:

- Firstly, we build **Support Vector Machines** (SVM) with the help of **SVM^light** [12]. We start by stemming tokens and filtering function words to generate a classical VSM (see above). In the case of the German corpus we perform a full lemmatization. Based on the resulting VSM we derive more than 16,000 lexical features to represent input documents. The next step is to learn a separate SVM for each of the 10 target categories (cf. Table 6 and 10). This is done in a one-against-all setting by training linear kernels. The evaluation is performed by means of the leave-one-out method. Note that we decided to use a linear kernel to save training effort — this leaves plenty room for improving our approach.

- Secondly, we start from the same VSM to perform a latent semantic analysis (LSA) in conjunction with \(k\)-means clustering, \(k = 10\) (cf. Table 7 and 11). Note that we average the results of \(k\)-means clustering over 10 repetitions, while we select 300 main components within the *single value decomposition* step of the LSA. Other than the SVM-based classifier, this approach does not need any training, but only knowledge about the number of target classes. The idea of performing LSA is to come up with a reduced feature matrix which ideally represents more explicitly inherent semantic relations of lexical features. See [14] for a thorough description of this approach.

- Thirdly, we vary the latter approach by including frequent phrases as additional features (cf. Table 7 and 11).

- Fourthly, we compute **SEQ**-based classifiers as described in Section 4.4 (cf. Table 8 and 12).

- Fifthly and finally, we implement the Wikipedia-based classifier of Section 4.5 which instead of a classical search engine uses the Wikipedia to derive information about the semantic relatedness of terms (cf. Table 9 and 13).

To evaluate these approaches we compute the \(F\)-measure (or \(F\)-score) as a standard evaluation technique in IR [39].\(^3\) The results of this experiment are discussed subsequently.

6 Discussion

Looking at the results of classifying English documents by their OAI data (cf. Tables 6–9), it is evident that Wikipedia- and LSA-based classifiers produce the lowest \(F\)-scores (\(F = .407\) and \(F = .469\), respectively). Of course, this finding is conditioned by the scenario under consideration. Notwithstanding this result, we observe that by enhancing the vector space model with the help of frequent phrases we raise the \(F\)-score to .5. Further, we see that the SEQ-based classifier (cf. Table 8) outperforms the Wikipedia- and LSA-based approach with an \(F\)-score of .626. This value is much above the outcome of the corresponding random baseline scenario (cf. Table 7), that is, .171. In other words, in the case of more

\(^3\) The \(F\)-score of a classification is the harmonic mean of its precision and recall.
than 60% of the input documents the DDC main class is correctly assigned by exploring OAI metadata snippets — this is much better than a random classifier which is informed about the extension of the target classes. Not surprisingly, the SVM-based classifier performs best (cf. Table 6). With an overall $F$-score of .761, SVMs provide an adequate DDC-related method to classify documents based on their OAI metadata. That is, in up to 75% of the documents, the classification is correct, however at the cost of a much higher training effort than induced by the less expensive SEQ-based classifier.

Regarding the German corpus data (cf. Table 10–13), a poorer performance is observed by means of all five different classifiers. The SEQ-based classifier performs now with an overall $F$-score of .275 — the worst result among all candidates (which is only under-run by the random baseline scenario). In contrast to the English case, the Wikipedia-based classifier (cf. Table 13) performs now much better, that is, with an $F$-score of .477. Once more, we observe that the
Table 8. Results of SEQ-based classification of English OAI data.

<table>
<thead>
<tr>
<th>DDC</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>.516</td>
<td>.874</td>
<td>.649</td>
</tr>
<tr>
<td>100</td>
<td>.691</td>
<td>.739</td>
<td>.714</td>
</tr>
<tr>
<td>200</td>
<td>.730</td>
<td>.852</td>
<td>.786</td>
</tr>
<tr>
<td>300</td>
<td>.529</td>
<td>.446</td>
<td>.484</td>
</tr>
<tr>
<td>400</td>
<td>.645</td>
<td>.848</td>
<td>.733</td>
</tr>
<tr>
<td>500</td>
<td>.786</td>
<td>.740</td>
<td>.762</td>
</tr>
<tr>
<td>600</td>
<td>.878</td>
<td>.360</td>
<td>.511</td>
</tr>
<tr>
<td>700</td>
<td>.833</td>
<td>.500</td>
<td>.585</td>
</tr>
<tr>
<td>800</td>
<td>.706</td>
<td>.444</td>
<td>.593</td>
</tr>
<tr>
<td>900</td>
<td>.888</td>
<td>.444</td>
<td>.593</td>
</tr>
<tr>
<td>Overall</td>
<td>.720</td>
<td>.611</td>
<td>.626</td>
</tr>
</tbody>
</table>

Table 9. Results of Wikipedia-based classification of English OAI data.

<table>
<thead>
<tr>
<th>DDC</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>.911</td>
<td>.720</td>
<td>.804</td>
</tr>
<tr>
<td>100</td>
<td>.691</td>
<td>.380</td>
<td>.490</td>
</tr>
<tr>
<td>200</td>
<td>.682</td>
<td>.580</td>
<td>.627</td>
</tr>
<tr>
<td>300</td>
<td>.564</td>
<td>.310</td>
<td>.400</td>
</tr>
<tr>
<td>400</td>
<td>.825</td>
<td>.470</td>
<td>.599</td>
</tr>
<tr>
<td>500</td>
<td>.694</td>
<td>.430</td>
<td>.531</td>
</tr>
<tr>
<td>600</td>
<td>.509</td>
<td>.290</td>
<td>.369</td>
</tr>
<tr>
<td>700</td>
<td>.778</td>
<td>.700</td>
<td>.737</td>
</tr>
<tr>
<td>800</td>
<td>.605</td>
<td>.460</td>
<td>.523</td>
</tr>
<tr>
<td>900</td>
<td>.625</td>
<td>.300</td>
<td>.405</td>
</tr>
<tr>
<td>Overall</td>
<td>.689</td>
<td>.464</td>
<td>.549</td>
</tr>
</tbody>
</table>

Table 10. Results of SVM-based classification of German OAI data.

<table>
<thead>
<tr>
<th>Baseline Term Phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-Score:</td>
</tr>
<tr>
<td>.148 .398 .468</td>
</tr>
</tbody>
</table>

Table 11. F-measure results of term- and phrase-based LSA of German OAI data. The baseline classification is performed by a random mapping of input objects to 10 target classes where the classifier is informed about the correct extension of the target classes.

LSA-based classifier is enhanced by including frequent phrases into the selection of features. Further, the SVM-based classifiers outperform again all other approaches by an overall F-score of .549. In a nutshell, while in the case of the English data, the Wikipedia-based classifier is the poorest performer, the German data is a test case where this classifier is the second-best approach.

The general decline from the English to the German test results can be explained by the descriptive gap induced by a loss of tokens used to build the metadata under consideration. In the case of the English corpus, an OAI summary consists on average of 84 tokens, while in the German texts there are only about 15 — this is a loss of more than 75% of the descriptive units used to induce lexical features. As an example, take the following OAI of a German document:

1 Der eingebildete Kranken = (Le Malade imaginaire) 840 ; 792 ; 11 ;
2 dk01 Repertoir des Herzoglich Meiningen’schen Hof-Theaters ; 6
<table>
<thead>
<tr>
<th>DDC</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>.277</td>
<td>.520</td>
<td>.361</td>
</tr>
<tr>
<td>100</td>
<td>.279</td>
<td>.410</td>
<td>.332</td>
</tr>
<tr>
<td>200</td>
<td>.315</td>
<td>.640</td>
<td>.422</td>
</tr>
<tr>
<td>300</td>
<td>.285</td>
<td>.370</td>
<td>.322</td>
</tr>
<tr>
<td>400</td>
<td>.226</td>
<td>.260</td>
<td>.242</td>
</tr>
<tr>
<td>500</td>
<td>.500</td>
<td>.180</td>
<td>.265</td>
</tr>
<tr>
<td>600</td>
<td>.394</td>
<td>.130</td>
<td>.195</td>
</tr>
<tr>
<td>700</td>
<td>.276</td>
<td>.080</td>
<td>.124</td>
</tr>
<tr>
<td>800</td>
<td>.370</td>
<td>.272</td>
<td>.314</td>
</tr>
<tr>
<td>900</td>
<td>.282</td>
<td>.129</td>
<td>.177</td>
</tr>
<tr>
<td>Overall</td>
<td>.320</td>
<td>.299</td>
<td>.275</td>
</tr>
</tbody>
</table>

**Table 12.** Results of SEQ-based classification of German OAI data.

<table>
<thead>
<tr>
<th>DDC</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>.546</td>
<td>.650</td>
<td>.594</td>
</tr>
<tr>
<td>100</td>
<td>.422</td>
<td>.430</td>
<td>.426</td>
</tr>
<tr>
<td>200</td>
<td>.737</td>
<td>.730</td>
<td>.734</td>
</tr>
<tr>
<td>300</td>
<td>.410</td>
<td>.160</td>
<td>.230</td>
</tr>
<tr>
<td>400</td>
<td>.738</td>
<td>.620</td>
<td>.674</td>
</tr>
<tr>
<td>500</td>
<td>.514</td>
<td>.370</td>
<td>.430</td>
</tr>
<tr>
<td>600</td>
<td>.431</td>
<td>.690</td>
<td>.531</td>
</tr>
<tr>
<td>700</td>
<td>.735</td>
<td>.360</td>
<td>.483</td>
</tr>
<tr>
<td>800</td>
<td>.332</td>
<td>.760</td>
<td>.462</td>
</tr>
<tr>
<td>900</td>
<td>.319</td>
<td>.150</td>
<td>.204</td>
</tr>
<tr>
<td>Overall</td>
<td>.518</td>
<td>.492</td>
<td>.477</td>
</tr>
</tbody>
</table>

**Table 13.** Results of Wikipedia-based classification of German OAI data.

Evidently, such an input is a little on the short side which makes it too difficult to classify the corresponding document correctly. Of course, the document belongs to literature (i.e., DDC class 800). However, lemmata as *Kranke(r)*/invalid associate unrelated content which may disturb the classifier.\(^4\)

From this point of view, the classification results produced using the German corpus are remarkably high, especially in the case of the Wikipedia-based classifier, while the search engine-based classifier fails because of the loss of lexical descriptors in the German corpus. Moreover, we may also conclude that the minimum number of lexical descriptors should range above 80 per OAI summary. Of course, to give an exact statement about this number requires further research.

The results of our DDC-related document classification are in a sense promising, in that we may think of DDC classifiers which solely explore document metadata. However, we also learn that these metadata should be extensive enough to prevent misclassifications — *beyond what given DDC metadata snippets provide*. Thus, control of the quality of OAI metadata becomes crucial when it comes to building metadata-based classifiers based on OTMs as approached by the SEQ- and the Wikipedia based classifiers of Section 4. Actually, we can imply an optimal metadata extension much below the range of full documents, but also above the extension of the German document samples collected in our corpus. Search-engine- and Wikipedia-based classifiers are promising candidates to realize this approach. But first and foremost, SVM-based classifiers produce the best result — *at the cost of significantly increased training time and resources*. Although the SEQ-based classifier does not need training, it produces a total amount of \(719 \times 300 = 215,700\) search queries — this is a secondary source of expense to be

\(^4\) Note that “Der eingebildete Kranke” is the German title of Molière’s “The Hypochondriac”.
considered carefully. Note that the text resources of OAI snippets are research papers, presentations or dissertations with up to 100 pages and more. Therefore, we can expect that the title, summary and keywords of a document will provide sufficient information to classify this document supposed that this feature resource is extended by its abstract. Of course, this is a feasible extension, so we are optimistic about improving the existing range of DDC classifiers — in support of Hypothesis H1.

7 Conclusion

In this article we introduced and evaluated several content-related classifiers used in digital libraries. The classifiers explore OAI metadata as a source of document representation and focus on the DDC as a system of target categories. To overcome problems of data sparseness, our approach explores web-based resources such as, e.g., the Wikipedia, to enhance feature extraction and selection. Our evaluation proves the potential of using OAI-metadata-based document representations. However, the $F$-scores of our approach which are below 90% indicate plenty room for improvement. Finally, our evaluation emphasizes the need for controlling the quality of this metadata and for enhancing it by additional document-related information as provided, for example, by abstracts.

Acknowledgement

We gratefully acknowledge financial support of the German Research Foundation (DFG) through the EC 277 Cognitive Interaction Technology, the Research Group 437 Text Technological Information Modeling and the DFG-LIS-Project P2P-Agents for Thematic Structuring and Search Optimization in Digital Libraries at Bielefeld University. We also thank the Bielefeld University Library which kindly provided the test data used in this article.
Bibliography


[37] Benno Stein, Sven Meyer zu Eissen, and Nedim Lipka. Web genre analysis: Use cases, retrieval models, and implementation issues. In Mehler et al. [20].


