Computational Intelligence, Volume 33, Number 000, 2017

Automated Classification of Content Components in Technical Communication

JAN OEVERMANN
University of Bremen, Bremen, Germany &
Karlsruhe University of Applied Sciences, Karlsruhe, Germany

WOLFGANG ZIEGLER
Karlsruhe University of Applied Sciences, Karlsruhe, Germany

Automated classification is usually not adjusted to specialized domains, due to a lack of suitable data collections and insufficient characterization of the domain-specific content and its effect on the classification process. This work describes an approach for the automated multi-class classification of content components used in technical communication based on the vector space model. We show that differences in form and substance of content components require an adaption of document-based classification methods and validate our assumptions with multiple real-world data sets in two languages.

As a result we propose general adaptions on feature selection and token weighting as well as new ideas for the measurement of classifier confidence and the semantic weighting of XML-based training data. We introduce several potential applications of our method and provide a prototypical implementation. Our contribution beyond the state of the art is a dedicated procedure model for the automated classification of content components in technical communication which outperforms current document-centered or domain-agnostic approaches.

Key words: Content Management; Machine Learning; Technical Communication; Text Classification.

1. INTRODUCTION

Large and complex documents used in technical communication are often composed of smaller building blocks, called *content components*[1] (Andersen, 2011). This enables referenced reuse of components across and within different documents and cost efficient translation in cases where only a subset of a document is changed (Soto et al., 2015). Examples for these document types are any kind of technical information (manuals, reports, educational material) but also standards documents, patents and some specifications types. Content components can resemble, but are not limited to, subsections of a document and are in most cases conceptually self-contained.

Component content management systems (CCMS) are a popular way to create, manage and assemble content components, especially for the creation of multi-authored documents (Grahlmann et al., 2010). In most cases content is written and stored in semantically structured XML-based *information models*[2] (Di Iorio et al., 2012) and manually enriched with metadata, such as predefined classification models, in order to identify content components for retrieval and distribution (Drewer and Ziegler, 2011). Advanced CCMS use classifications for the automated assembly of information products (e.g. printed manuals). Information models can be either native to the CCMS, in-house development of the company or standardized (such as *DocBook* (OASIS, 2008) or *PI-Mod* (Ziegler, 2011)).

Address correspondence to Jan Oevermann and Wolfgang Ziegler, Karlsruhe University of Applied Sciences, Faculty for Information Management and Media, Moltkestraße 30, 76133 Karlsruhe, Germany

[1] In other literature and commercial applications content components are also referred to as "topics", "modules" or "content modules" (Rockley et al., 2003; Drewer and Ziegler, 2011).

[2] Information models are often referred to by their technical representation: DTDs (document type definitions) or schemas.

For technical documentation, hierarchical *PI classification taxonomies*[3] or related methods are a popular and established framework to classify content components for these tasks (Ziegler and Beier, 2015). The assignment of classes is usually done by technical writers at the time of development and is based on experience and editorial guidelines. However, classifying large amounts of content manually (e.g. when migrating legacy data) is time consuming and prone to error. To the best of our knowledge, there are currently no specialized tools or specific methods available to automate this task, which focus on the characteristics of content components. This is in contrast to the demands of the technical documentation sector, which rely on targeted and reliable information retrieval, e.g. for service technicians or technical personnel. Classifying metadata can be used to integrate content components in more dynamic scenarios, where information is aggregated or filtered automatically and utilized in faceted search (Zheng et al., 2013; Broughton, 2006). This is reflected in the rising popularity of Content Delivery Portals (CDP) which provide users with a metadata-based way to access modular information (Ziegler and Beier, 2015). Classifying metadata can also serve as a basis for advanced information access methods, where classification criteria is preset through QR codes, RFID chips or user location. Due to the amount of legacy content in most companies, automated classification is necessary to provide these information services. Recently published standards for the exchange of technical documentation integrate these ideas with Linked Data technologies (European Association for Technical Communication - tekom e.V., 2017).

Classification based on the vector space model (VSM) is a well known (Manning and Schütze, 1999; Sebastiani, 2002) and performance efficient (Le and Mikolov, 2014) way to do such bulk classification. However, most applications are optimized for entire documents and not only parts thereof regarding feature extraction and weighting. In addition, most implementations focus on plain text and do not recognize semantic structures (Di Iorio et al., 2014), e.g. in XML-based training data, which is widely used in component content management and can contain additional meta information about the content.

In this work we want to present an approach which considers these particularities of content components and adjusts standard vector space classification for better accuracy in this specialized task. We introduce a new weighting method and feature extraction for content components, first approaches to confidence measuring and semantic weighting of XML elements with special meaning in information models. Furthermore we investigate correlations regarding the classification framework used for the content, number and type of classes as well as language and size of components. Our experiments serve as basis for more research on this domain-specific application of machine learning methods.

2. RELATED WORK

This article is an extended version of work published by Oevermann and Ziegler (2016). We expand our previous work by validating our results with eight more classification tasks, including an additional data set and an added language for one of the sets. An additional weighting technique and other token combinations are used in experiments to get a more detailed comparison. We present new insights on semantic weighting, confidence score reliability and stemming. We added current and future applications and provide a working prototype including the source code of the implementation.

Characteristics of text in component content management applications were discussed,

[3] Where *PI* is a reference to **P**roduct and **I**nformation, the two dimensions in which information can be classified intrinsically (Drewer and Ziegler, 2011).

among others, by Bailie and Huset (2015), Andersen (2011), Drewer and Ziegler (2011), Grahlmann et al. (2010) and Rockley et al. (2003).

Effective feature extraction methods for text classification were discussed by Biricik et al. 2012 and 2009. The authors introduce a new method called “Abstract Feature Extraction” motivated by TF-IDF and TF-ICF, which significantly improves accuracy across different classifiers. The TF-IDF-CF method we base our token weighting on was introduced and tested by Liu and Yang (2012). More weighting schemes are discussed and compared by Ko (2012) and Lan et al. (2005).

Domain-specific approaches for automated classification are discussed by Golub (2006) focusing on web documents. The author sees the lack of available document collections as one of the reasons for missing classification research on certain domains. One instance of a detailed domain-centered approach is the work by Caldas et al. (2002). The authors analyze automated classification methods and their applications for the domain of construction project documents. Similarities of their work and our work are the availability of predefined classification frameworks and the focus on process automation of the classification task.

Information retrieval for XML-based documents is extensively covered in the literature. In work by Lalmas (2009) several ranking methods for elements in XML-based documents are discussed. The author comes to the conclusion that hierarchical context (e.g. ancestor elements) matters most when comparing retrieval scoring strategies for XML elements. We describe a similar approach with class hierarchies in section 8.3. Kotsakis (2002) presents a ranking approach, that combines TF-IDF with a coefficient of the structural element positioning. Disadvantages of these methods are, that for each single path in the collection a coefficient has to be chosen. This approach was not suitable for our data sets. Di Iorio et al. (2014) describe how textual XML documents are based on multiple element patterns, which can have different levels of relevance (e.g. junk structures vs. representational structures). Their conclusions verify our hypothesis, that a weighting of elements in XML-based content is reasonable (especially in the case of so called surrogate elements).

Research on utilizing machine learning methods and similarity measures in the field of technical documentation was recently done by Soto et al. (2015). Their work describes methods to aid technical writers in reusing content components with automated text similarity measures. Their method could be combined with ours, to verify if components identified for reuse have matching classes assigned.

3. METHODOLOGY

After characterizing several important properties of component contents based on industry best practices and international standards, we make conclusions about the effects of these characteristics on classification tasks based on VSM. We verify our adaption in a test set-up with 11 classification tasks on four real-world data sets consisting of about 7000 manually classified content components in two languages.

3.1. Scope

Although the experiments in this work were solely performed with content obtained from technical documentation, its results and conclusions are applicable to a wider scope of documents, which meet the characteristics defined in section 4, such as certain international standards, patents or modularized business documents (e.g. specifications).

3.2. Test data

Our test data consists of different kinds of technical information and was provided by companies from different engineering sectors for research purposes (see table 1). Due to the company-confidential nature of the contents, the data sets cannot be made public. The content language is either German or English. One data set (D) contains about 80 content components available in both languages. All content was written by professionals (technical writers or subject matter experts) according to editorial standards, therefore, a certain quality can be expected of the data available (see also section 4.6).

In comparison to previous experiments (Oevermann and Ziegler, 2016) we could enlarge the overall corpus of content components, add one more language and an entire new company-provided set.

TABLE 1. Training and test sets

Set	Info. model[4]	Language	Units	$\frac{Words}{Unit}$	Classification system	Classes
A	syst.	en-US	1087	515	2-level information type	10 / 26
B	open	de-DE	4186	87	2-level information type	6 / 22
					1-level product type	28
C	corp.	de-DE	663	180	1-level information type	11
					1-level product type	22
D	open	de-DE	584	51	2-level information type	8 / 14
		en-GB	1070	57	2-level information type	8 / 17

All data sets are XML-based and have classifications that follow a PI classification taxonomy for information types with one or two levels. Two data sets have additional product-related classifications. The number and labels of classes and the average size of components differ from company to company (see also section 4.2). As an example for typical class labels, we list classes contained in set C (10 examples each, translated from German):

Intrinsic information classes (information type)

Maintenance, Assembly/Disassembly, Operation, Diagnosis, Emergency operation, Product information, Safety information, Troubleshooting, Transport, Preface/Introduction.

Intrinsic product classes (product type)

Drive group, Work equipment, Work hydraulics, Equipment/Options, Brake system, Electrical system, Hydraulic components, Cooling system, Steering system, Lubricating system.

Sets B and D are structured according to the open source *PI-Mod* information model (Ziegler, 2011), set A was stored as a CCMS-specific variant of semantic HTML and set C was provided in an custom information model used by the company.

[4]The XML-based information model (IM) the content was created in. syst.: CCMS-specific IM, open: standardized open source IM, corp.: in-house developed corporate IM

3.3. Preprocessing

In a preprocessing step, all plain text from components was extracted and unnecessary white-space, digits, special characters and punctuation were removed. Features were extracted as a combination of single words and word groups (as described in section 5.1) and then weighted with the TF-ICF-CF method described in section 5.3. No stemming was applied on words for reasons discussed in 5.2.

3.4. Test set-up

In supervised learning we build a $n \times c$ token-by-class matrix $M = \{w_{ij}\}$ for a set of distinct classes C. For each token i we calculate the class-specific *semantic weight* w_{ij} for class j (Ko, 2012).[5] Each class is therefore represented by a prototypical class vector $\vec{p_j}$, which contains the characteristic token distribution of the class across all content components in the training data (cf. section 5.3).

A content component for classification is represented as a vector $\vec{m} = (w_1, w_2, ...w_n)$ where n is the number of tokens chosen as features of the component and w is the weight of the tokens. The similarity is calculated for each component-class combination and the confidence score for the prediction (closest class) is derived (see section 6).

All classification tasks in this work are multi-class problems. Our set-up is based on a vector space model, instead of on a more sophisticated method (such as neural networks), for performance reasons. For the classifier we chose simple *cosine similarity* (Manning and Schütze, 1999) instead of support vector machines or naïve Bayes due to the high numbers of features and the heterogeneous size and distribution of classes (Colas et al., 2007).

The same set of parameters and configurations was used for all classification tasks independent of the language of the data set. Compared to previous results (Oevermann and Ziegler, 2016) no semantic weighting for specific XML-elements was used to allow for a better comparison of results (see section 5.4).

3.5. Measurement

For 10-fold cross validation we randomly divided the test data into a training set and a test set (9:1) (Kohavi, 1995). Results are measured as *mean accuracy* (Sokolova and Lapalme, 2009) across all classes in the set. Variance between the results of the individual cross validation tests are indicated as mean squared error (MSE).[6]

4. CHARACTERISTICS

In the following sections, we outline characteristics of content components, which can effect classification tasks based on the vector space model.

4.1. Content types

The specialized domain of technical communication covers the writing and structuring of user manuals. Content and document sections contained therein are constrained in many ways by standards and regulatory rules. One of the most important regulations stipulates predefined content types in the sequence of traditional chapter structures for manuals and

[5] Weights are normalized per class j as $\frac{w_{ij}}{\sum w_{ij}}$

[6] MSE calculated as $s_{N-1} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})^2}$.

interactive electronic technical documentation (IEC 82079-1, 2012). These sections are resembled by content components in CCMS. The corresponding content types follow the lifecycle of engineering products (2006/42/EC, 2006).

This covers, for example, information about transportation, installation and adjustment of machinery, or instructions on how to use, maintain and dispose a product. Additional technical data, advice on safety issues and conceptual or other descriptive information (e.g. configuration, layout and functionality of the product) must also be included. The relevant regulations for a product, therefore, demand certain sets of content types for the corresponding technical documentation. These sets are often an essential part of specialized information models. Well known examples are manuals for military and avionic vehicles or for medical devices (S1000D, 2012; ATA iSpec 2200, 2014; GHTF/SG1/N70, 2011).

4.2. Classification models

For CCM applications, predefined content types usually translate into a distinct set of *intrinsic information classes*, while documents (which can contain several content types) are classified with *extrinsic information classes*, such as “service manual” or “user guide”. The same kind of classification can be applied to product-related properties. In this case *intrinsic product classes* are used to associate content with the assembly group or part of the product that is described (e.g. “hydraulic pump” or “cooling unit”). *Extrinsic product classes* relate to the products (model, series), for which the content is valid.

This metadata-driven approach for content management is defined as *PI classification method* by Ziegler (Drewer and Ziegler, 2011) and was developed for the classification of content components. Usually, PI classification models are defined as taxonomies and describe at least a two dimensional information space. Further extensions of the metadata model can take more complex relations into account, e.g. in-between content components in ontology-based approaches. Since most CCMS in technical communication are restricted to simple taxonomies or even lists, we focus on these.

Each content component has to have distinct coordinates in the information space of intrinsic product and information classes. Technical writers assign these classes to content components at the time of development and have to follow the corresponding rules for text preparation. Main use cases of the method are an efficient retrieval in CCM or CDP applications, the automated aggregation of documents and classification-based cross-referencing.

In the course of this paper, we focus mainly on *intrinsic information classes*, because of their direct connection to the linguistic properties and underlying information models of content components. In addition, we test the same classification methods on *intrinsic product classes* to evaluate if these require different adjustments to the classification process. Extrinsic classes are not covered by this work, as they are, in most cases, multi-label problems and can be solved in other ways. For example, the assignment of *extrinsic product classes* may be solved through named entity recognition.

4.3. Standardized patterns

Standardized grammatical patterns are used within content components to increase consistency and reusability across multiple documents. Especially technical manuals have to be concise and unambiguous, due to their normative nature and recurring structural and grammatical patterns are used to improve reading comprehensibility, e.g. for safety advises (ANSI Z535.6, 2006). These rules are often resembled in editorial guidelines or style guides (IEC 82079-1, 2012), which remind technical writers to abstain from the use of synonyms, ambiguity, direct speech, filler words, sentiments or empty phrases. Some companies even utilize controlled languages like *Simplified Technical English* to further restrict grammar and vocabulary (Kincaid et al., 1991).

Standardizing patterns can also reduce translation costs when used in combination with *translation management systems* (Allen, 1999) and improve readability. For example, grammatical patterns differ in style, whether they depict instructive or descriptive content. This helps readers to differentiate between different types of content, such as descriptions, tasks or safety advises in manuals.

Content components of one information class often demand only one kind or one specific combination of grammatical patterns. XML-based information models, such as DITA (OASIS, 2010), DocBook (OASIS, 2008) or PI-Mod (Ziegler, 2011), incorporate this as special XML-elements for semantically different content components (cf. default DITA topic types: concept, task, reference, etc.). Controlled language checker software or authoring guidelines can help enforce these grammatical and syntactical rules depending on the content type.

4.4. Specific terminology

Terminology and choice of words used in technical and normative documents is often highly specific to the subject the content describes and is, in some cases, strictly controlled within the principles of *terminology work* and enforced by *terminology management system* (ISO 704, 2009; ISO 26162, 2012). Characteristic terms are often precise technical expressions which are usually unique to the engineering sector the content belongs to.

As occasionally parts of technical communication material are also used for marketing purposes, some companies explicitly mention the full brand/model combination with every occurrence of the product name for better brand recognition. This leads to highly characteristic word distributions in content components which has advantages for classification performance but restricts the use of the trained model for content aside from the scope trained in (either the company or the branch of the company).

4.5. Size of components

The actual size of content components depends on several factors, such as strategic decisions, product complexity or software features of the CCMS (Drewer and Ziegler, 2011). Component properties have been analyzed systematically for various data sets from companies and results range from small content fragments with just a few words up to components including several hundreds or thousands of words. For one example, a corpus examined by Oberle and Ziegler (2012) had an average component size of about 150 words, whereas the usual size of a document was approximately 12,000 words.

Fragments are usually included within other content components, but can also be manually classified within CCMS (for example safety advices in manuals). Small size content fragments are used, for example, in more complex reuse scenarios within variant management functionality of CCM applications (Rockley et al., 2003).

The data sets examined for this work had average word counts between 51 and 515 words per content component (cf. table 1). The size of components is, therefore, significantly smaller than that of typical documents (approx. 1:75, which also equals the average number of content components in one document). This results in fewer features per unit which can be evaluated by prediction algorithms in comparison to document classification and a high variance in size for different data sets (cf. table 1).

4.6. Training and test data

Companies using component content management in combination with a well defined classification model already have high quality training material at hand which is suitable for supervised learning. Content was classified manually by experts and written in a controlled manner according to editorial guidelines. Standardized information models can also provide further information about semantic properties and functions of parts of the text, as well as, semantically enriched HTML (see section 5.4). However, for some parts, the highly technical nature of the content can have a negative impact on classification performance (e.g. for tables, legends or lists).

Data for automated classification can either be structured but unclassified content components from CCMS and other sources (e.g. before classification models were introduced in a company) or unstructured and unclassified PDF documents or other file formats used for archiving. Especially these legacy files play an important role in technical documentation because manufacturers have a legal obligation to retain documents for several years after product deployment (e.g. in the European Union the mandatory duty to preserve documentation for machinery is 10 years (2006/42/EC, 2006)).

This discrepancy results in potential differences between training and test data regarding format, structure and quality, which a domain-specific classification procedure has to take into account.

4.7. Quality assurance

Due to high legal standards and safety implications that adhere to technical communication material, a proper quality assurance is mandatory before publishing (IEC 82079-1, 2012; ISO 9001, 2008). Especially in the European Union all necessary technical documentation for machinery is considered as integral part of the product (2006/42/EC, 2006). The correctness and completeness of published documents is, therefore, crucial for the integrity of the whole product. Because some CCMS rely on classifications of content components for an automated composition of documents, the classification algorithm is a possible vulnerability for product integrity. These requirements entail the need for a measurable confidence score which can be used as a threshold for quality assurance in cases where the classification results may be unreliable.

5. IMPLICATIONS

In the following section we derive implications for supervised learning and automated classification of content components from characteristics presented in the previous section and verify them with our test data (cf. table 1).

5.1. Feature selection

Standardized terminology and grammatical word patterns decrease the total number of distinct words and word combinations in technical documentation in comparison to other text types. This is generally preferable in text classification, as it reduces the usual high dimensionality of the feature space (Caldas et al., 2002). As content components are also much smaller than documents (cf. section 4.5), the amount of features for representation of an object is further reduced. However, most content components in technical communication have both distinct single words and recognizable word patterns as important characteristics of their information or product related classification (see explanation in sections 4.3 & 4.4). Although an optimal feature selection depends on the specific characteristics of a data set, a combination of single words and word patterns works best on across diverse data sets (cf. table 2 for w_{ij} = TF-ICF-CF).

Our results confirm the assumption that a combination of n-grams (where n is the number of words per group) is in most cases the preferable method for representing content

TABLE 2. Accuracy for different n-grams as tokens (classification task: information type level 1), where n is the number of words per group for w_{ij} = TF-ICF-CF. Best result per set in bold.

n	Set A (en) [%]	Set B (de) [%]	Set C (de) [%]	Set D (en) [%]	Average [%]
1	86.7	78.5	75.9	75.5	79.2
2	91.7	85.4	81.7	82.1	85.2
3	**92.5**	85.9	73.6	75.7	81.9
4	92.1	83.7	67.9	73.6	79.3
$\{1,2\}$	90.1	83.5	79.7	80.7	83.5
$\{2,3\}$	91.9	**87.0**	81.1	82.4	85.6
$\{1,2,3\}$	91.6	85.3	**82.6**	**83.6**	**85.8**

components (cf. Table 2 for w_{ij} = TF-ICF-CF). Taking the MSE of cross validation tests into account (between 1-3% on all tests) it demonstrates, that the best average accuracy is achieved by combined word patterns selected as features ($n = \{1,2\}$ and $n = \{1,2,3\}$). Because a high number of features can negatively impact computing performance, $n = 2$ is the best preference for industrial applications. Processing time for classification can be greatly reduced while maintaining good accuracy. There was no significant correlation between the language of the content and the optimal value for n.

5.2. Stemming

Applying a list-based stemming for German-language test data did not improve classification accuracy and in some cases even decreased classifier performance. This behavior can be traced back to the use of word patterns as features, which can convey important grammatical meaning (e.g. verb conjugations when classifying information types). As a consequence we did not use any stemming on words.

5.3. Feature weighting

There are several ways to assign contextual weight to a feature with TF-IDF as the best known method (Caldas et al., 2002; Ko, 2012; Lan et al., 2005; Liu and Yang, 2012). TF-IDF, mostly used in document retrieval, combines overall *term frequency* with the *inverse document frequency* (e.g. in Baeza-Yates and Ribeiro-Neto (1999)), which serves as a indicator on how unique a specific term i is for a document n:

$$w_{ij} = tf_{ij} \cdot \log(\frac{N}{n_i}) \tag{1}$$

There are several TF-IDF variants, which for example apply smoothing to account for cases where term frequency is zero (Liu and Yang, 2012). We will refer to this variant as TF-IDF$_{\text{smooth}}$:

$$w_{ij} = \log(1 + tf_{ij}) \cdot \log(\frac{1+N}{n_i}) \tag{2}$$

To improve accuracy of document categorization, TF-IDF has been extended to TF-IDF-CF, which considers in-class characteristics of features (Liu and Yang, 2012):

$$w_{ij} = \log(1 + tf_{ij}) \cdot \log(\frac{1+N}{n_i}) \cdot \frac{tf_{ij}}{C_j} \tag{3}$$

However, in CCM the reference size of one unit is a content component and not a document.

TABLE 3. Accuracy for different weighting methods (classification task: information type level 1) for $n = \{1, 2, 3\}$. Best result per set in bold.

w_{ij}	Set A (en) [%]	Set B (de) [%]	Set C (de) [%]	Set D (en) [%]	Average [%]
TF-IDF	25.3	50.5	25.2	47.6	37.2
TF-IDF$_{\text{smooth}}$	63.5	39.8	25.2	65.4	48.5
TF-IDF-CF	85.8	69.2	72.6	69.4	74.3
TF-ICF-CF	**91.6**	**85.3**	**82.6**	**83.6**	**85.8**

Therefore, document-based weighting is not necessarily suitable for component classification tasks. Due to the nature of our training data, from which we can derive overall *token frequency* tf_i as well as *in-class frequency* cf_{ij} and *inverse class frequency* icf_{ij}, we adapted TF-IDF-CF to utilize *inverse class frequency* (ICF) to differentiate between classes instead of IDF. For a set of distinct classes C with classes j and tokens i weight w_{ij} calculated with TF-ICF-CF is:

$$w_{ij} = \log\left(1 + tf_i\right) \cdot \log\left(1 + \frac{|C|}{tf_i}\right) \cdot \frac{tf_{ij}}{C_j} \tag{4}$$

Our results confirm that TF-ICF-CF performs best as weighting method on our data compared to other document-oriented schemes (cf. table 3 for $n = \{1, 2, 3\}$).

An additional advantage of the proposed TF-ICF-CF metric is the independence from document units (in this case: content components) when calculating the weights, which leads to a very performance efficient training phase, that only has linearly dependencies on the number of classes and the number of extracted features. This is possible, because in PI classification frameworks a content component is always an instance of one intrinsic class or class combination.

The TF-ICF-CF measure is therefore suitable for applications which focus on mutual characteristics of classes and don't rely on information about single units within a class (such as in information retrieval). Examples for applications of TF-ICF-CF include supervised learning (as shown in this work) and quality measures for classification systems (e.g. class distribution, missing classes, average class distance).

5.4. Semantic quantifiers

Semantic information about text structure of content components is usually available in training data in the form of XML elements, their attributes or their *meta-structure* (Di Iorio et al., 2012). However, this additional information is usually missing in test data (as for example in legacy documents, such as PDF), which makes a direct comparison difficult. To circumvent this, it is possible to artificially increase the term frequency tf_i with a quantifier q for tokens that have special semantic meanings in one class (e.g. headings, emphases, results), so that in supervised learning tf_i is extended to:

$$tf_{iq} = tf_i * q \quad \text{for } q > 0 \tag{5}$$

Results from previous work (Oevermann and Ziegler, 2016) show that for $2 < q < 5$, classification accuracy can improve up to 10% ($q = 2.5$). This can be traced back to tackling a well-known problem in high-dimensional feature selection for text classification: lack of good predictive features, which discriminate a class (Forman, 2004). This effect is also known as the *siren pitfall*.

However, quality and choice of semantic structures for quantification heavily influence the benefits of semantic qualifiers. Elements for semantic weighting can only be chosen by hand at the moment, leading to a difficult and biased direct comparison between data sets with different information models. We tried to automate this selection based on two hypotheses: (1) Selection of elements which are unique for one class and are therefore more relevant to that class and (2) applying the TF-ICF-CF weighting method to all elements and choosing the ones with the highest weight per class. Both attempts did not significantly improve classification results or even decreased accuracy in cross validation due to overfitting.

5.5. Confidence scoring

Based on the reasons discussed in section 4.7, it is highly desirable to measure confidence of classification results for integrating automated classification into an editorial workflow. While in retrieval scenarios, like filtering in a CDP, wrong or missing classification is inconvenient (*recall* is important), it can be crucial for automated publishing processes (*precision* is important).

There are several methods for comparing per-class classification scores s_c, such as the *softmax function* or the *standard deviation*, however neither of them suited our need for a reliable quality assurance measure. We therefore, decided to use a simple ratio-based score (see eq. 6).

$$r = \frac{s_1 - s_2}{s_1 - s_n} \tag{6}$$

We base our confidence score r on the presence of single outliers (high confidence) or close runner-ups (low confidence). Per-class classification scores s_c for n classes c are sorted from high (1) to low (n). r is then expressed as ratio of first to second and first to last classification choice.

6. RESULTS & DISCUSSION

We tested our assumptions with 11 different classification tasks based on 4 data sets (A-D) which are outlined in table1. General results are listed in table 4, details can be found in tables 2, 3 and 5. The different tasks result from company-specific PI classification models of the data and vary considerably in their characteristics.

6.1. General results

The best results were achieved with intrinsic level-1 classifications of information types ($91.6\% \pm 1.7$ on 10 classes for set A) and product types ($82.5\% \pm 2.1$ on 28 classes for set B). With set-specific feature selection accuracy could be further improved up to $92.5\% \pm 2.3$ on 10 classes for set A. Level-2 results of information classes vary between 74.8% (D) and 87.8% (A). For this scenarios accuracy could be increased by incorporating fallback mechanisms as described in section 8.3.

The outcome of our experiments show, that a VSM-based classification process can be highly viable in the presented use cases (cf. section 7). All results are based on a generalized parameter set (*zero configuration*) and unmodified production data exported from CCMS, which is of major importance for a real-world application of our method. For certain cases, we could show with our data, that product classification problems can be solved with the same methods as information type classification. Experiments on more data sets have to be carried out to confirm this observation.

Based on our limited data we are not able to draw final conclusions about the effect of outer factors (information model, set size, component size, language, classification type and number of classes) on accuracy results. Consistent with our subjective estimation we found,

TABLE 4. Accuracy for different classification tasks for $n = \{1, 2, 3\}$ and w_{ij} = TF-ICF-CF.

Set	Language	Classification task	Classes	Accuracy [%]	MSE
A	en-US	information type (level 1)	10	91.6	1.7
		information type (level 2)	26	87.8	3.6
B	de-DE	information type (level 1)	6	85.3	2.5
		information type (level 2)	22	78.0	2.3
		product type (level 1)	28	82.5	2.1
C	de-DE	information type (level 1)	11	82.6	3.1
		product type (level 1)	22	74.5	7.1
D	de-DE	information type (level 1)	8	78.4	6.8
		information type (level 2)	14	80.7	4.8
	en-GB	information type (level 1)	8	83.6	3.7
		information type (level 2)	17	74.8	4.1

that the most important factor for good classification performance is high quality content. This includes a well defined classification model, correctly performed manual classification and text written in a standardized manner according to writing guidelines. To measure these quality aspects of content in a non-subjective way is a topic for future research.

6.2. Correlations

We found measurable correlations[7] between:

(1) the size of the training data and the MSE of cross validation tests ($\rho = -0.6$)
(2) the average size ($\frac{words}{unit}$) of content components and the classification accuracy ($\rho = 0.7$)

Although not significant for our data, we expect to find correlations between the number of classes and classification accuracy on a wider scope of data collections, because the probability of correct classification decreases with an increasing number of classes. Within single data sets this is already observable, when comparing accuracy results between levels 1 and 2 on information classes. One anomaly, for level-2 classification having better accuracy than level-1 for the German data of set D, lies within the MSE.

6.3. Selection and weighting of features

Results in tables 2 and 3 show, that the selection and weighting of features can be adjusted towards the characteristics of content components.

Due to the standardized nature of texts in technical communication we can show, that a combination of word groups is the best way to represent content components (with bigrams as an performance-efficient alternative).

For weighting these features, the TF-ICF-CF method could significantly improve classification results over document-oriented approaches (cf. table 3).

[7]Correlation measured as Pearson correlation coefficient: $\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y}$

6.4. Classification confidence

To test the reliability of a quality control which is based on a confidence threshold, we calculate confidence scores on cross validation and isolate content components which have a wrong classification but a high confidence score ($r > 0.7$).

TABLE 5. Fraction of wrongly classified content components (F) where confidence score $r > 0.7$ (classification task: information type level 1) for $n = \{1, 2, 3\}$ and w_{ij} = TF-ICF-CF.

	Set A (en) [%]	Set B (de) [%]	Set C (de) [%]	Set D (en) [%]	Average [%]
$F_{r>0.7}/F$	4.49	1.59	9.49	4.76	5.08

Our results show, that a well chosen threshold (here: $r > 0.7$) enables fully automated workflows, where automatically classified content components with a high confidence score can be processed without further manual control while keeping error rate low.

6.5. Limitations

Due to the lack of available research on automated classification in the field of technical communication there is no baseline for cross-comparison of results. Additional experiments on the same data with alternative machine learning methods for text classification need to be conducted in order to get a more general evaluation.

7. APPLICATIONS

The following sections give a short overview of potential applications for the automated classification of content components.

7.1. Authoring assistance

Authors, who create content in CCMS, usually set the class of the content when they start writing. In some cases the content changes over time or the author chooses the wrong class. This can lead to problems in identifying the component at a later stage of the lifecycle. Before storing the newly written content, automated classification can act as additional quality assurance in the background by comparing the manually assigned class with results of the automated classification (Oevermann, 2016a). If they differ, the author should be advised to double check the assigned class (e.g. in form of a software-triggered confirm dialog box).

7.2. CMS data migration

With the introduction of a CCMS, companies often start using classification models (e.g. PI classification) to take advantage of more advanced features, such as document aggregation or retrieval functions. Furthermore, it can be observed that the implementation of a CCMS can effect writing styles of authors (Bailie and Huset, 2015). To migrate existing (structured) content to the new CCMS it is also necessary to add classification to legacy content, which is a time consuming task. A possible solution to reduce manual work is to select a representative fraction of the corpus (e.g. 500 - 1000 content components) as training set and classify the remaining content in an automated way. Based on a confidence threshold, technical writers can then review content components for which the assigned classification might be wrong.

7.3. Unstructured documents

Another application based on component-based classification is the (re-)segmentation of unstructured documents, such as legacy PDFs, for the use in information retrieval, as described by Oevermann (2016b).

Text is extracted from a PDF document and split into words. The set of extracted words W is grouped into arbitrary text chunks $C = \{c_1, ..., c_n\}$, where $c_i \subset W$. The size of these chunks is based on the average word count of a content component: a (which can be derived from the training data, cf. table 1). Text chunks are distributed across the document via an offset $r \in \mathbb{N}$ with $r \leqslant a$. This offset defines the starting position for each chunk. Therefore, a text chunk c_i at position i can be defined as (for $i > 1$): $c_i = \{W_{(i-1)*r}, W_{(i-1)*r+1}, ..., W_{(i-1)*r+a}\}$ (Oevermann, 2016b). In the following step all text chunks are sequentially classified with the methods described in this paper and the confidence of each classification is calculated (cf. section 5.5). Through clustering of similar classified text chunks, segments of similar semantic meaning (according to the intrinsic class) can be reconstructed (e.g. sections of a specific information type or on a specific part on the product). Boundaries between sections can be observed, when the confidence of the classifier falls to a local minimum or below a defined threshold. This allows us to narrow down page ranges and regions within pages of a specific classification in a way that is completely independent from any formatting information contained in the PDF file and can also work with plain text obtained from an OCR-based preprocessing.

7.4. Structured search

Content which is classified can be made available for faceted search which allows users to narrow down full-text searches by e.g. type of information.[8] Filtering by class is also a common use case for content delivery portals, where some filters can even be set automatically (e.g. derived from service orders) to prepare information. Further applications include auto suggestions per classification or context-aware searches.

8. OUTLOOK

In upcoming work we will extend our research further to other data sets and focus on unstructured documents as source for classification. We plan to refine our models to include grammatical patterns and use alternative classification techniques. In future research we also want to include alternative feature extraction and weighting methods, such as AFE (Biricik et al., 2009).

8.1. Language

Globalized companies write content components in one source language and translate them into several target languages. This makes it possible to measure classification accuracy for the same content across different languages. Results could answer questions about whether some languages are more suitable for classification based on statistical NLP than others or if classification accuracy decreases after translation. First experiments on feature selection (see section 5.1) could not find any correlations but have to be repeated with a larger collection of content components in different languages.

If the language of the content does not affect accuracy, the classification results could be used to rank translations or translation vendors. The underlying hypothesis for such a ranking

[8] As an example: a service technician might only need maintenance information for his assigned task

states that a good translation does not change classification accuracy. Therefore, translation quality could be tested in an automated way, where significant decreases in accuracy are used as indicators for poor translations.

8.2. Linguistic features and word order

At the moment features for classification are only obtained through statistical methods and do not incorporate linguistic features such as the verb form or the inflection of a noun. This information could be used to improve classification results, especially when classifying information types in technical communication where grammatical patterns often convey the type of content (e. g. instructional or descriptive). Another important aspect of grammatical patterns is word order (Le and Mikolov, 2014). In the current implementation this information is only preserved within n-grams but not in the context of content components. Using the position of a word or word pattern within an content component as an additional feature could improve accuracy.

8.3. Taxonomy fallback

In the current implementation, the classification process is unaware of existing hierarchical relations between classes and considers every level of the taxonomy as a separate set of classes. With this method, accuracy decreases to subordinate classes (cf. table 4). Reasons can be found, for example, in the increasing semantic and syntactical similarity of neighboring classes, making it harder for the classifier to distinguish between them.

Especially for use cases in information retrieval it may, therefore, be desirable to have fallback mechanisms in place. They can resort to a taxonomic parent class if classifier confidence is below a certain threshold and ensure that *recall* for class-based filtering of content components stays high in lower levels of the classification model. This behavior of trading accuracy for usability is also known as *graceful degradation* (Menychtas and Konstanteli, 2012).

Another way to use taxonomic knowledge for improving accuracy is to include classification scores of higher-level classes in cases, where two classes have similar cosine measures but different parent classes. If confidence is higher on the superordinate level, this can used to distinguish subordinate classes.

8.4. Quality measures for classification models

The quality of the underlying classification model heavily influences performance of automated classification. For example, if classes are ambiguous, training data may be skewed due to wrong manual classification or similar instances belonging to different classes. In classification scenarios based on the vector space model, such as ours, this can often be observed as abnormal distribution of class vectors. Ambiguous classes tend to have similar directions while other classes may be missing, if the classifier regularly places them between the same two classes. In future research we want to develop a model, that can predict such anomalies and help to measure the quality of a classification model based on training data.

Furthermore, well defined and distinct classification models and good classification by technical writers also should result in a close to 100% accuracy rate when training and validating with the same data set (self validation). Though overfitting of the trained model is generally not desirable, this behavior can be used in other ways. Self validation can be utilized to measure general quality of classification or the overall classification model. We observed in our tests that classification errors in self validation can be a strong indicator of wrong manual classification or an ambiguous classification model. Generating reports, which highlight content components with classification mismatches in self validation can help to spot wrongly classified objects. In future research we want to extend these reports with information about the presumed reason of the mismatch (e.g. either problems with the classifications system or with the object).

9. CONCLUSIONS

Content components used in technical communication have special characteristics which entail the need for a domain-specific classification method. Our results show that a tailored procedure model for this content type can improve accuracy in classification tasks in comparison to more general or document-centered approaches. As shown in this article, there are multiple real-world scenarios where automated classification is applicable and necessary, especially by enabling advanced information access methods in the industrial sector (e.g. for service technicians). PI classification models provide a suitable framework for these applications and can be incorporated in machine learning scenarios, as shown in our experiments. First results show that different classification tasks can be solved by the introduced method. In comparison to previous work we could expand the scope to product-related classifications, which we could successfully cover with our method.

We identified several areas of domain-specific adjustments and made proposals for improving classifier performance. The improvements include the combination of word group combinations ($n = \{1, 2\}$ and $n = \{1, 2, 3\}$) as features for classification and a modified token weighting scheme for in-class characteristics (TF-ICF-CF). Through additional experiments we could also show, which features can serve as alternative for performance-critical training tasks ($n = 2$). We made recommendations for the use of stemming, hierarchical classifications, semantic quantifiers and a confidence measuring on cosine similarity classifier results. Furthermore we discussed and tested classification behavior for different class types and levels within the PI classification method. Finally, we presented several applications of our method and proposed topics for future research on this subject. Our adjustments have shown significant improvements compared to document-oriented or more general methods and are a first step towards an automated classification of content components in technical communication.

ACKNOWLEDGMENTS

We would like to thank Christoph Lüth (University of Bremen) for insightful discussions, Stephan Steurer for generous support and Heike Berumen for thorough proofreading. The tests in this work were performed with data collections kindly provided by companies that want to support research on this subject.

SUPPLEMENTARY MATERIALS

We implemented a platform-independent framework for the proposed methods to verify the application-ready approach. The framework is designed to allow for easy extension, configuration and usage. Therefore, it is possible to add additional functionality such as similarity measures for content components, analyses for unstructured content or exports of results in various data formats. A browser-based demo[9] and corresponding source code[10] are publicly available.

[9] `http://coin.fastclass.de`
[10] `https://github.com/j-oe/coin-demo`

REFERENCES

2006/42/EC. 2006. Machinery directive of the European Parliament and of the Council.

ALLEN, JEFFREY. 1999. Adapting the Concept of "Translation Memory" to "Authoring Memory" for a Controlled Language Writing Environment. *In* Proceedings of the 21. International Conference on Translating and the Computer, Volume 10-11, London. http://mt-archive.info/Aslib-1999-Allen.pdf.

ANDERSEN, REBEKKA. 2011. Component Content Management - Shaping the Discourse through Innovation, Diffusion, Research and Reciprocity. Technical Communication Quarterly, **20**(4):384–411.

ANSI Z535.6. 2006. American National Standard for Product Safety Information in Product Manuals, Instructions, and Other Collateral Materials.

ATA ISPEC 2200. 2014. Information Standards for Aviation Maintenance. https://publications.airlines.org/CommerceProductDetail.aspx?Product=185.

BAEZA-YATES, R., and BERTHIER RIBEIRO-NETO. 1999. Modern Information Retrieval. Addison-Wesley, New York : Harlow, England. ISBN 978-0-201-39829-8.

BAILIE, RAHEL ANNE, and JEFFREY HUSET. 2015. The Effect of CMS Technology on Writing Styles and Processes: Two Case Studies. IEEE Transactions on Professional Communication, **58**(3):309–327. ISSN 0361-1434, 1558-1500. . http://ieeexplore.ieee.org/document/7393884/.

BIRICIK, GÖKSEL, BANU DIRI, and AHMET COŞKUN SÖNMEZ. 2009. A new method for attribute extraction with application on text classification. IEEE. ISBN 978-1-4244-3429-9. pp. 1–4. . http://ieeexplore.ieee.org/document/5379479/.

BIRICIK, GÖKSEL, BANU DIRI, and AHMET COŞKUN SÖNMEZ. 2012. Abstract feature extraction for text classification. Turkish Journal of Electrical Engineering & Computer Sciences, **20**(Sup. 1):1137–1159.

BROUGHTON, VANDA. 2006. The need for a faceted classification as the basis of all methods of information retrieval. Aslib Proceedings: New Information Perspectives, **58**(1/2):49–72. ISSN 0001-253X. . http://www.emeraldinsight.com/doi/10.1108/00012530610648671.

CALDAS, CARLOS H., LUCIO SOIBELMAN, and JIAWEI HAN. 2002. Automated Classification of Construction Project Documents. Journal of Computing in Civil Engineering, **16**(4):234–243. ISSN 0887-3801, 1943-5487. . http://ascelibrary.org/doi/10.1061/%28ASCE%290887-3801%282002%2916%3A4%28234%29.

COLAS, FABRICE, PAVEL PACLK, JOOST N. KOK, and PAVEL BRAZDIL. 2007. Does SVM Really Scale Up to Large Bag of Words Feature Spaces? *In* Advances in Intelligent Data Analysis VII. *Edited by* M. R. Berthold, J. Shawe-Taylor, and N. Lavra, Volume 4723. Springer Berlin Heidelberg, pp. 296–307. Berlin, Heidelberg. ISBN 978-3-540-74824-3 978-3-540-74825-0. http://link.springer.com/10.1007/978-3-540-74825-0_27.

DI IORIO, ANGELO, SILVIO PERONI, FRANCESCO POGGI, and FABIO VITALI. 2012. A First Approach to the Automatic Recognition of Structural Patterns in XML Documents. *In* Proceedings of the 2012 ACM Symposium on Document Engineering, DocEng '12, ACM, New York, NY, USA. ISBN 978-1-4503-1116-8. pp. 85–94. . http://doi.acm.org/10.1145/2361354.2361374.

DI IORIO, ANGELO, SILVIO PERONI, FRANCESCO POGGI, and FABIO VITALI. 2014. Dealing with structural patterns of XML documents. Journal of the Association for Information Science and Technology, **65**(9):1884–1900.

DREWER, PETRA, and WOLFGANG ZIEGLER. 2011. Technische Dokumentation. Vogel, Würzburg.

EUROPEAN ASSOCIATION FOR TECHNICAL COMMUNICATION - TEKOM E.V. 2017. iiRDS Specification - intelligent information Request and Delivery Standard - First Public Working Draft, 03 April 2017. https://iirds.tekom.de.

FORMAN, GEORGE. 2004. A Pitfall and Solution in Multi-class Feature Selection for Text Classification. *In* Proceedings of the Twenty-first International Conference on Machine Learning, ICML '04, ACM, New York, NY, USA. ISBN 1-58113-838-5. pp. 38–. . http://doi.acm.org/10.1145/1015330.1015356.

GHTF/SG1/N70. 2011. Label and Instructions for Use for Medical Devices.

GOLUB, KORALJKA. 2006. Automated subject classification of textual web documents. Journal of Documentation, **62**(3):350–371. ISSN 0022-0418. . http://www.emeraldinsight.com/doi/10.1108/00220410610666501.

GRAHLMANN, KNUT, COCKY HILHORST, SANDER VAN AMERONGEN, REMKO HELMS, and SJAAK BRINKKEMPER. 2010. Impacts of implementing Enterprise Content Management Systems. *In* ECIS 2010 Proceedings, Pretoria, South Africa, p. 103.

IEC 82079-1. 2012. Preparation of Instructions for Use - Structuring, Content and Presentation.

ISO 26162. 2012. Systems to manage terminology, knowledge and content - Design, implementation and maintenance of terminology management systems.

ISO 704. 2009. Terminology work - Principles and methods.

ISO 9001. 2008. Quality management systems - Requirements.

KINCAID, J PETER, CALLIOPI D KINCAID, JD KNIFFIN, MARGARET THOMAS, and SHEAU LANG. 1991. Intelligent Authoring Aids for Technical Instructional Materials Written in Controlled English. Journal of Artificial Intelligence in Education, **2**(3):77.

KO, YOUNGJOONG. 2012. A Study of Term Weighting Schemes Using Class Information for Text Classification. *In* SIGIR12, ACM.

KOHAVI, RON. 1995. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *In* IJCAI'95 Proceedings of the 14th international joint conference on Artificial intelligence, Volume 2, Morgan Kaufmann, Montreal, Canada, pp. 1137–1145.

KOTSAKIS, EVANGELOS. 2002. Structured Information Retrieval in XML Documents. *In* Proceedings of the 2002 ACM Symposium on Applied Computing, SAC '02, ACM, New York, NY, USA. ISBN 1-58113-445-2. pp. 663–667. . http://doi.acm.org/10.1145/508791.508919.

LALMAS, MOUNIA. 2009. XML information retrieval. *In* Encyclopedia of Library and Information Sciences.

LAN, MAN, CHEW-LIM TAN, HWEE-BOON LOW, and SUNG SUNG. 2005. A Comprehensive Comparative Study on Term Weighting Schemes for Text Categorization with Support Vector Machines. *In* 14th International World Wide Web Conference, ACM.

LE, QUOC, and TOMAS MIKOLOV. 2014. Distributed Representations of Sentences and Documents.

LIU, MINGYONG, and JIANGANG YANG. 2012. An improvement of TFIDF weighting in text categorization. *In* International Conference on Computer Technology and Science (ICCTS 2012).

MANNING, CHRISTOPHER D., and HINRICH SCHÜTZE. 1999. Foundations of statistical natural language processing. MIT Press, Cambridge, Mass.

MENYCHTAS, ANDREAS, and KLEOPATRA G. KONSTANTELI. 2012. Fault Detection and Recovery Mechanisms and Techniques for Service Oriented Infrastructures. *In* Achieving Real-Time in Distributed Computing: From Grids to Clouds. IGI Global, pp. 10.4018/978–1–60960–827–9.ch014. Hershey, PA.

OASIS. 2008. The DocBook Schema, Committee Draft 5.0. http://www.docbook.org/specs/docbook-5.0-spec-cd-03.html.

OASIS. 2010. DITA Version 1.2 Specification. http://docs.oasis-open.org/dita/v1.2/spec/DITA1.2-spec.html.

OBERLE, CLAUDIA, and WOLFGANG ZIEGLER. 2012. Content Intelligence for Content Management Systems. *In* tcworld, (12). http://www.tcworld.info/rss/article/content-intelligence-for-content-management-systems/.

OEVERMANN, JAN. 2016a. Intelligente Klassifizierung von technischen Inhalten – Automatisierung und Anwendungspotenziale. *In* Proceedings of tekom Conference 2016, tcworld, Stuttgart, Germany.

OEVERMANN, JAN. 2016b. Reconstructing Semantic Structures in Technical Documentation with Vector Space Classification. *In* Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems - SEMANTiCS2016 and the 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS'16), Volume 1695, CEUR-WS, Leipzig, Germany. http://ceur-ws.org/Vol-1695/paper5r1.pdf.

OEVERMANN, JAN, and WOLFGANG ZIEGLER. 2016. Automated Intrinsic Text Classification for Component Content Management Applications in Technical Communication. *In* Proceedings of the 2016 ACM Symposium on Document Engineering, ACM Press, Vienna, Austria. ISBN 978-1-4503-4438-8. pp. 95–98. . http://doi.acm.org/10.1145/2960811.2967153.

ROCKLEY, ANN, PAMELA KOSTUR, and STEVE MANNING. 2003. Managing Enterprise Content: A Unified Content Strategy. New Riders, Berkley, CA.

S1000D. 2012. Issue 4.1: International specification for technical publications using a common source database. http://public.s1000d.org/Downloads/Documents/Issue4.1/S1000D%20Issue%204.1.zip.

SEBASTIANI, FABRIZIO. 2002. Machine learning in automated text categorization. ACM Computing Surveys, **34**(1):1–47. ISSN 03600300. . http://portal.acm.org/citation.cfm?doid=505282.505283.

SOKOLOVA, MARINA, and GUY LAPALME. 2009. A systematic analysis of performance measures for classification tasks. Information Processing & Management, **45**(4):427–437.

SOTO, AXEL J., ABIDALRAHMAN MOHAMMAD, ANDREW ALBERT, AMINUL ISLAM, EVANGELOS MILIOS, MICHAEL DOYLE, ROSANE MINGHIM, and MARIA CRISTINA FERREIRA DE OLIVEIRA. 2015. Similarity-Based Support for Text Reuse in Technical Writing. *In* Proceedings of the 2015 ACM Symposium on Document Engineering, DocEng '15, ACM, New York, NY, USA. ISBN 978-1-4503-3307-8. pp. 97–106. . `http://doi.acm.org/10.1145/2682571.2797068`.

ZHENG, BWEIJUNL, WEI ZHANG, and XIAOYU FU BOQIN FENG. 2013. A survey of faceted search. Journal of Web engineering, **12**(1&2):041–064.

ZIEGLER, WOLFGANG. 2011. PI-Mod: An information model for plant construction and mechanical engineering (and others). `http://pi-mod.de/index.php?lang=en`.

ZIEGLER, WOLFGANG, and HEIKO BEIER. 2015. Content delivery portals: The future of modular content. *In* tcworld e-magazine, (02/2015).