Automated Intrinsic Text Classification for Component Content Management Applications in Technical Communication

Jan Oevermann
University of Bremen &
Karlsruhe University of Applied Sciences
76133 Karlsruhe, Germany
jan.oevermann@hs-karlsruhe.de

Wolfgang Ziegler
Karlsruhe University of Applied Sciences
76133 Karlsruhe, Germany
wolfgang.ziegler@hs-karlsruhe.de

ABSTRACT

Classification models are used in component content management to identify content components for retrieval, reuse and distribution. Intrinsic metadata, such as the assigned information class, play an important role in these tasks. With the increasing demand for efficient classification of content components, the sector of technical documentation needs mechanisms that allow for an automation of such tasks. Vector space model based approaches can lead to sufficient results, while maintaining good performance, but they must be adapted to the peculiarities that characterize modular technical documents.

In this paper we will present domain specific differences, as well as characteristics, that are special to the field of technical documentation and derive methods to adapt widespread classification and retrieval techniques for these tasks. We verify our approach with data provided from companies in the sector of manufacturing and mechanical engineering and use it for supervised learning and automated classification.

Keywords

Technical Documentation; Content Management; Vector Space Model; Machine Learning; Text Classification

1. INTRODUCTION

Complex documents, such as technical product documentation required in industrial engineering, are mostly composed of small *content components*[1] that allow for referenced reuse and cost efficient translation [21]. XML-based component content management systems (CCMS) provide a professional environment to create and assemble these components.

CCMS are often enhanced by classification methods in order to identify content components for retrieval and distribution [5]. For example the assignment of information classes is usually done manually by technical writers at the time of creation and is based on experience and editorial guidelines. However, for large amounts of content, (e.g. migrating legacy data) this method is extremely time consuming. There are currently no tools or specific methods available for automating this task focusing on characteristics of technical product documentation.

Vector space classification is, in general, an efficient way to do such bulk classifications but is often optimized towards whole documents and not parts thereof. In addition, the method usually does not recognize semantic structures which are widely used in component content management (CCM). Therefore, CCM content is in most cases ideal training data due to its semantic richness, consistent style of writing and the XML-based data format.

In our approach we want to consider these peculiarities of technical documentation and adjust standard vector space classification to utilize them for better accuracy in automated classification tasks.

Table 1: Training and test sets

Set	Sector	Units	$\frac{\text{words}}{\text{unit}}$	Classes
A	mechanical eng.	570	173	11
B	mechanical eng.	278	41	10
C	manufacturing eng.	3947	97	22

2. METHODOLOGY

At first we characterize important properties of component content management based on industry best practices and international standards. We then make assumptions about the effects on classification tasks and verify them in a test set-up with three different real-world data sets (about 4,800 manually classified content components). All test data was provided by companies in the sectors of manufacturing and mechanical engineering and is in German.

In preprocessing, text from components was extracted and unnecessary punctuation and XML syntax removed (for use of semantics see section 4.3). The multi-class test set-up was based on a vector space model (VSM), instead of more sophisticated methods (such as *Neural Networks*), for performance reasons. 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. The value w_i represents the *semantic weight* of token i [11]. In supervised learning we built a $n \times c$ token-by-class matrix $M = \{w_{ij}\}$ for a set of distinct classes C. As classifier we use *cosine similarity* [14].

For cross validation we randomly divided the test data into a training set and a validation set (4:1).

[1] In other literature and commercial applications *content components* are also referred to as *topics*, *modules* or *content modules* [5, 19].

DocEng '16, September 12 - 16, 2016, Vienna, Austria

ISBN 978-1-4503-4438-8/16/09...$15.00

DOI: http://dx.doi.org/10.1145/2960811.2967153

3. CHARACTERISTICS OF CCM

The following sections outline characteristics of CCM that are different from other content types and which are relevant for classification tasks based on vector space models.

3.1 Classification models

In the field of technical communication, manuals and document sections contained therein, are constrained in many ways by standards and regulatory rules. One of the most important regulations states predefined content types in the sequence of traditional chapter structures of manuals and of interactive electronic technical documentation [7]. Well known examples are manuals of military and avionic vehicles or of medical devices [6, 18, 20]. For content component management applications, this usually translates into distinct sets of information classes, which then depend on specific business domains. Content components have to be created according to the predefined information classes and are, therefore, instances of one intrinsic information class. Consequences for the use of terminology and other editorial guidelines are outlined in the following sections.

A metadata-driven approach for defining content components is defined as *PI Classification* method in [5]. In this model *intrinsic* metadata is coupled with product components by corresponding product classes (P) and include the required set of information classes (I). Usually, PI classification models are defined as taxonomies and describe an, at least, two dimensional information space. Each content component has to have distinct coordinates in the space of intrinsic product and information classes. Technical writers have to assign content to a unique class and have to follow the corresponding rules for content creation.

In this framework, there are additional *extrinsic* PI classes describing the intended or actual use of components in end-products and final document types (which can usually be coupled to named-entity recognition). In the course of this paper, we focus mainly on the *intrinsic* information classes. The most common starting point for defining information classifications are the distinct sets of descriptive vs. procedural content classes. Descriptive content includes, for example, set-up of machines, process or functional descriptions of hard and software or introductory sections. Procedural content covers all types of task-oriented information like installation, how-to-use instructions, maintenance and repair or disposal of products. In general, procedural content and the corresponding taxonomy of information classes is organized according to the product life cycle. The *intrinsic* information classes used in industrial CCMS applications build up a well-known set of classifications and are a starting point for our approach to automated text classification.

3.2 Standardized patterns

Due to their normative nature, technical documents have to be concise and unambiguous. This is often resembled in editorial guidelines or *style guides* [7], which remind technical writers to abstain from the use of synonyms, ambiguity, direct speech, filler words, sentiments or empty phrases. Instead standardized grammatical patterns are used within content components to increase consistency and reusability across multiple documents. This decreases translation costs when used in combination with translation management systems (TMS) and improves reading comprehension for users. These patterns differ in style, whether they depict instructive or descriptive content. This helps readers to differentiate, for example, between tasks, concepts or *embedded safety messages* [2]. Content components of one information class often contain only one kind or one specific combination of grammatical patterns and word classes (e.g. only imperative verbs in instructions).

XML-based information models, such as DITA [16], DOCBOOK [15] or PI-MOD [23], reflect this with semantic content components, as for example “descriptive”, “task” or “concept”.

3.3 Specific terminology

Terminology and overall choice of words used in technical documents is often highly specific to the company that manufactures the product and is strictly controlled within the principles of *terminology work* and enforced by *terminology management system* [8, 9]. Terms for describing tasks and concepts are mostly precise technical expressions that are usually unique to the engineering sector, to which the product belongs (e.g. printing presses or construction machinery).

For better brand recognition among customers, some companies also explicitly mention the full brand/model combination with every occurrence of the product. This leads to very characteristic word distributions in content components that are often unique for one company or even one branch of a company.

3.4 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. Component properties have been analyzed systematically for various 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 corpus in [17], the average component size was about 150 words, whereas the usual size of a document was approximately 12,000 words (German language).

Fragments are usually included within other content components, but can also be manually classified within CCMS. One can find that small size content fragments are used, for example, in more complex reuse scenarios within variant management functionality of CCM applications [19, 22].

The data examined for this paper had average word counts per content component or fragment of respectively 173, 97 and 41 words (cf. Table 1). The size of components is, therefore, significantly smaller than that of typical documents (approx. 1:75). This results in fewer features per unit which can be evaluated by prediction algorithms in comparison to document classification.

3.5 Training and validation data

Companies, which are using component content management in combination with a well defined classification model already have high quality training material at hand that 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 [4]. However, for some parts, the technical nature of the content has a negative impact on classification performance (e.g. for tables, legends or lists).

Validation data can either be unclassified content components (from sources other than the CCMS) or unstructured and unclassified PDF documents or other file formats used for archiving. This results in potential differences between training and validation data regarding format and structure of the content.

3.6 Quality assurance

Due to high safety standards and legal implications that adhere to technical documentation, a proper quality assurance is mandatory before publishing [7, 10]. Especially in the European Union all necessary technical documentation for machinery is considered

Table 2: Accuracy for different n-grams as tokens

n	Set A [%]	Set B [%]	Set C [%]	Avg. [%]
1	79.26	77.78	67.13	74.72
2	80.49	73.91	76.88	77.09
3	78.75	68.42	76.29	74.49
$\{1,2\}$	82.14	80.43	73.70	78,76
$\{1,2,3\}$	**90.48**	**85.36**	**78.12**	**84.65**

as part of the product [1]. 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 the automated composition of documents, the classification algorithm is a possible vulnerability for product integrity. This entails the need for manual control in cases where the classification algorithm is not confident in its results.

4. IMPLICATIONS

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

4.1 Feature selection

Standardized wording and grammatical patterns decrease the total number of distinct words and word combinations in technical documentation in comparison to other text types (cf. section 3.2 & 3.3). This is generally preferable in text classification because it reduces the usually high dimensionality of the feature space [3]. As content components are also much smaller than documents (cf. section 3.4), the number of features for representing an object for classification is further reduced by a great amount.

However, most content components in technical communication have both distinct single words and recognizable word patterns as important characteristics of their information classes. This means using single terms or n-grams (e.g. bigrams or trigrams) as exclusive features is not optimal. Our results confirm the assumption that a combination of n-grams of different n is in most cases the preferable method for representing content components (cf. Table 2 for $q = 2.5$ and w_{ij} = TF-ICF-CF).

4.2 Token weighting

There are several ways to assign semantic weight to a token with TF-IDF as the best known method [3, 11, 12, 13]. To improve accuracy in document categorization, TF-IDF has been extended to TF-IDF-CF, which considers in-class characteristics of tokens [13]. However, in CCM the reference size of one unit is a content component and not a document. Therefore, document-based weighting is not always suitable for 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* tcf_{ij} and *inverse class frequency* icf_{ij}, we adapted TF-IDF-CF to utilize inverse *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} is:

$$w_{ij} = \log(1 + tf_i) * \log(1 + \frac{|C|}{tf_i}) * \frac{tf_{ij}}{C_j} \quad (1)$$

Our results confirm that this method performs best as weighting method on our data compared to other schemes (cf. Table 3 for $q = 2.5$ and $n = \{1,2,3\}$).

Table 3: Accuracy for different weighting methods

w_{ij}	Set A [%]	Set B [%]	Set C [%]	Avg. [%]
TF-IDF	52.13	56.27	50.45	52.95
TF-IDF-CF	75.79	76.08	63.37	71.75
TF-ICF-CF	**90.45**	**85.36**	**78.12**	**84.65**

4.3 Semantic quantifiers

As shown in section 3.5, semantic information about the text structure of content components is usually available in training data but cannot be directly applied in classification due to the lack of reliable structure elements (as for example in legacy documents). To circumvent this, it is possible to artificially increase the term frequency tf_i with a quantifier q for tokens that have special semantic meaning in one specific class (e.g. function or setup descriptions, action sequences), so that in supervised learning tf_i is extended to:

$$tf_{iq} = tf_i * q \text{ for } q > 0 \quad (2)$$

Test results show that for q between 2 and 5, classification accuracy can be increased up to 10% ($q = 2.5$). However, quality and choice of semantic structures for quantification heavily influences the benefits of semantic qualifiers. Thus in future work we want to examine methods for compiling comprehensive lists of semantic structures which are relevant for token weighting and their corresponding quantifiers.

4.4 Confidence scoring

For reasons discussed in section 3.6, it must be possible to measure confidence of classification results in regard to content components which could belong to multiple classes. There are several methods for comparing per-class classification scores, such as the *softmax function* or the *standard deviation*, however neither of them suited our need for a reliable quality assurance measure.

$$p = \frac{s_1 - s_2}{s_1 - s_n} \quad (3)$$

We base our confidence score p 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). p is then expressed as ratio of first to second and first to last classification choice. After examining confidence scores on our test sets, we can see that only a small fraction ($0-3\%$) of content components are incorrectly classified and have high confidence scores ($p > 0.7$).

5. APPLICATIONS

In this section we want to give a short overview of potential applications for an automated classification of content components in technical communication.

Quality management.

Well defined classification models and good classification by technical writers should result in a close to 100% accuracy rate when training and validating with the same data set. This circumstance can be utilized to measure general quality of classification or the overall classification model. In our tests, we observed that classification errors in self-validation can be a strong indicator of wrong manual classification of a content component. Results for our data match our subjective rating with set A (97.21 %) and B (96.39 %) having high quality classification as opposed to Set C (89.03 %) with a more ambiguous classification model.

Data migration.

With the implementation 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. To migrate existing (structured) content to the system it is also necessary to have legacy content classified, which is a time consuming task. In this case, automated classification of content components, which can utilize newly composed and manually classified content as training data, is desired.

6. RELATED AND FUTURE WORK

Domain-specific classification and their applications for construction project documents were analyzed in [3]. Similarities of this work are the availability of predefined classification frameworks and the focus on automation of the classification task.

Research on utilizing text similarity measures to aid technical writers in reusing content components was presented in [21]. The results could be used to verify if components identified for reuse have matching classes assigned.

The TF-IDF-CF method we base our token weighting on was introduced and tested in [13]. More weighting schemes are discussed and compared in [11] and [12].

In future work we will extend our research further to other data sets and focus on unstructured documents as a source for classification. We examine optimization potentials for semantic quantifiers and confidence scoring. We plan to refine our models to include grammatical patterns with advanced NLP technologies (Part-of-speech tagging).

7. CONCLUSIONS

Component content management has different characteristics and requirements than default document classification but multiple real-world scenarios where automated classification is applicable and necessary. PI classification models provide a suitable framework for these applications.

We identified several areas of improvement and made proposals for adapting existing models for use in technical communication. The improvements include the combination of terms and *n*-grams as features for classification, a modified token weighting scheme for in-class characteristics, semantic quantifiers to leverage information present in training data and a first approach to reliable confidence scoring on cosine similarity classifier results.

Results of this paper are based on content components from specific engineering disciplines but can also be applied to other sectors (e.g. software documentation). Our adjustments have shown significant improvements over document-oriented classification techniques and are a good foundation for future research.

8. ACKNOWLEDGMENTS

We would like to thank Christoph Lüth (Univ. of Bremen) and Claudia Oberle (Karlsruhe UAS) for insightful discussions, Reilly Lorenz for thorough proofreading and Stephan Steurer for support.

9. REFERENCES

[1] 2006/42/EC. Machinery Directive of the European Parliament and of the Council, 2006.

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

[3] C. H. Caldas, L. Soibelman, and J. Han. Automated Classification of Construction Project Documents. *Journal of Computing in Civil Engineering*, 16(4):234–243, 2002.

[4] A. Di Iorio, S. Peroni, F. Poggi, and F. Vitali. 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, pages 85–94, New York, NY, USA, 2012. ACM.

[5] P. Drewer and W. Ziegler. *Technische Dokumentation*. Vogel, Würzburg (DE), 2011.

[6] GHTF/SG1/N70. Label and Instructions for Use for Medical Devices, 2011.

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

[8] ISO 26162. Systems to manage terminology, knowledge and content: Design, implementation and maintenance of terminology management systems, 2012.

[9] ISO 704. Terminology work – Principles and methods, 2009.

[10] ISO 9001. Quality management systems – Requirements, 2008.

[11] Y. Ko. A Study of Term Weighting Schemes Using Class Information for Text Classification. In *SIGIR 2012*. ACM, 2012.

[12] M. Lan, C.-L. Tan, H.-B. Low, and S. Sung. A Comprehensive Comparative Study on Term Weighting Schemes for Text Categorization with Support Vector Machines. In *14th International World Wide Web Conference (WWW 2005)*, 2005.

[13] M. Liu and J. Yang. An improvement of TFIDF weighting in text categorization. In *IPCSIT*, volume 47. IACSIT Press, Singapore, 2012.

[14] C. D. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT Press, Cambridge, 1999.

[15] OASIS. The DocBook Schema, CD 5.0, 2008.

[16] OASIS. DITA Version 1.2 Specification, 2010.

[17] C. Oberle and W. Ziegler. Content Intelligence for Content Management Systems. *tcworld e-magazine*, 2012(12), 2012.

[18] A. T. A. of America. ATA iSpec 2200: Information Standards for Aviation Maintenance, 2014.

[19] A. Rockley, P. Kostur, and S. Manning. *Managing Enterprise Content: A Unified Content Strategy*. New Riders, Berkley, 2003.

[20] S1000D. Issue 4.1: International specification for technical publications using a common source database. http://s1000d.org, 2012.

[21] A. J. Soto, A. Mohammad, A. Albert, A. Islam, E. Milios, M. Doyle, R. Minghim, and M. C. Ferreira de Oliveira. Similarity-Based Support for Text Reuse in Technical Writing. In *Proceedings of the 2015 ACM Symposium on Document Engineering*, DocEng '15, pages 97–106, New York, NY, USA, 2015. ACM.

[22] W. Ziegler. Variantenverwaltung in CMS – Fünf Methoden für die Feinarbeit. *technische kommunikation*, 27(3):40–44, 2005.

[23] W. Ziegler. PI-Mod: An information model for plant construction and mechanical engineering (and others). http://pi-mod.de/index.php?lang=en, 2011.