Alexander Mehler, Peter Geibel, Olga Pustylnikov

Structural Classifiers of Text Types: Towards a Novel Model of Text Representation

Texts can be distinguished in terms of their *content*, *function*, *structure* or *layout* (Brinker, 1992; Bateman et al., 2001; Joachims, 2002; Power et al., 2003). These reference points do not open necessarily orthogonal perspectives on text classification. As part of explorative data analysis, text classification aims at automatically dividing sets of textual objects into classes of maximum internal homogeneity and external heterogeneity. This paper deals with classifying texts into text types whose instances serve more or less homogeneous functions. Other than mainstream approaches, which rely on the vector space model (Sebastiani, 2002) or some of its descendants (Baeza-Yates and Ribeiro-Neto, 1999) and, thus, on *content-related lexical features*, we solely refer to *structural* differentiae. That is, we explore patterns of text structure as determinants of class membership. Our starting point are tree-like text representations which induce feature vectors and tree kernels. These kernels are utilized in supervised learning based on cross-validation as a method of model selection (Hastie et al., 2001) by example of a corpus of press communication. For a subset of categories we show that classification can be performed very well by structural differentia only.

1 Introduction

The basic idea of text classification is that the content, structure and shape of textual units vary, though not deterministically with the communicative situation or function they manifest. As this variation is not stochastic, we can build classes or types of textual units where members of the same class share class constitutive differentiae. Varying reference points of clarifying the ontological status of these differentiae lead to different notions of text types: If we focus on functional or situative criteria of class membership, we deal with so called *genres* (Martin, 1992; Ventola, 1987) or *registers* (Biber, 1995; Halliday and Hasan, 1989), respectively. Analogously, we speak of *hypertext sorts*, *digital genres* or *web genres* in the case of web documents (Santini, 2007). If we consider the composition of classes in terms of their *extension*, that is, from the point of view of enumerating their elements, we deal with *sorts* of documents – e.g. *text sorts* in the sense of Heinemann (Heinemann, 2000). If in contrast to this, class membership is defined in *intensional* terms, we deal with *text patterns* (Heinemann, 2000) or *superstructures* (van Dijk and Kintsch, 1983) as prototypical representations of class members, whose expectation-driven production/processing they support.

In this paper we focus on functionally demarcated text types for which we investigate to which degree class membership is manifested by structural differentia. The idea to predict the function of a text by the patterns it instantiates comes from the quantitative approach according to which distributional patterns vary with the text function (Biber, 1995). Starting from the weak contextual hypothesis (Miller and Charles, 1991) one might state that structural differences reflect functional ones while similar functions tend to be manifested by similarly structured texts. With a focus on registers, Biber (1995, p.59) puts this as follows: "preferred linguistic forms of a register are those that are best suited functionally to the situational demands of the variety [...]." As there is a many-to-many relation of structure and function (neither can we deterministically infer a unique function based on observing some text pattern, nor is the same function always manifested by the same pattern), learning text types by exploring text structures is a nontrivial task.

In the present paper, we focus on the logical document structure (Power et al., 2003) as a source of feature selection while we disregard layout and any content indicating lexical units of the texts to be classified. Since we focus on *text* types, we leave out hypertext and, especially, web documents.[1] Further, since we aim at modelling text *types*, we go beyond classical approaches which learn classifiers in order to enumerate class members without any effort in interpreting these classifiers as representations of text patterns. Rather, we perform our experiments as a preliminary step towards learning classifiers as representations of such patterns. As far as the classifiers being learnt allow deriving representations of patterns which, in turn, allow computing the similarity of texts with respect to these patterns, we contribute to a *prototype ontology* in the sense of Sowa (2000). For a remarkably large set of text types of press communication, we show that classification of their instances performs very well when disregarding any lexical features.

The paper is organized as follows: Section 2 discusses some related approaches; Section 3 presents two novel text representation models which are evaluated and discussed in Section 4. Finally, Section 5 concludes and prospects future work.

2 Related Work

In recent years, feature selection attracted many researchers in the field of classification. The aim is to find alternatives to the *bag-of-words* approach (Biber, 1995; Kessler et al., 1997; Karlgren, 1999; Lee and Myaeng, 2002; Wolters and Kirsten, 1999). Although lexical features are selective with respect to text content, this IR model generally disregards text structure. Now, modeling document structure comes into reach of machine learning (Dehmer, 2005). Some approaches even show that structural patterns allow to classify texts in the absence of any lexical information (Dehmer, 2005; Lindemann and Littig, 2006; Pustylnikov, 2006). Baayen et al. (1996) present a pioneering approaches in this field. They achieve good results in authorship attribution by focussing on frequencies

[1] For a related approach to web-documents cf. (Mehler, 2007; Mehler et al., 2006).

of constituent types (e.g., NP, VP). These observations indicate that authors have idiosyncratic syntactic signatures by which they can be classified. Further, Lindemann and Littig (2006) report good results in classifying web sites of different web genres (*blogs*, *personal*, and *academic homepages* as well as *online shops* and *corporate sites*). One of their resources of structural features is the link structure of the sites. Note that Mehler et al. (2006) have shown that such classifications are problematic when looking for web genres of a much higher resolution. The genres analyzed by Lindemann and Littig (2006) are in a sense general that one might expect their instances to be well separable in terms of their structure. This paper shows that such a structural classification is even possible for a wide range of more homogeneous rubrics of press communication.

Biber (1995) generally claims that "no single linguistic parameter is adequate in itself to capture the range of similarities and differences among spoken and written registers" and, thus, pleads for a multidimensional approach which takes a multitude of lexical and syntactical features into account. This is confirmed by Lewis (1992) who reports that compared to the bag-of-words approach there is no improvement if single features (e.g., *phrase pattern counts*) are taken into account. However, the extraction of a multitude of such features is time consuming and error-prone. Thus, an easy processable resource of expressive features is needed instead. The logical document structure and its quantitative characteristics is such a resource. It can be automatically computed for a wide range of genres and registers and is certainly easier accessible than either, e.g., rhetorical structure or syntactic structure. Evidence for this assumption comes from previous studies (Pustylnikov, 2006; Gleim et al., 2006) with respect to several registers and two languages (English and German). In this work we extend the structural framework by introducing *Quantitative Structure Analysis* (QSA) as a formal model of structural text representation models.

3 Text Representation Models Based on Structure-Sensitive Features

In recent experiments (Mehler et al., 2006), we have studied the selectivity of structural features in text classification. Our findings have shown that unsupervised learning of web document structures performs above the baseline scenario of random classification. As a complementary approach, we now tackle the question what "golden standard" can be achieved by using supervised methods. In order to do that, we investigate two structure-oriented text representation models as input to SVM-based machine learning:

1. *Quantitative structure analysis:* Our starting point is to represent texts by a set of quantitative features as a model of their structure. That is we build feature vectors whose coefficients do no longer stand for lexical units, but represent structural text characteristics. Section 3.1 presents a formal account of this approach which is inspired by Tuldava (1998) who clusters texts by means of simple quantitative characteristics. A further source of inspiration is synergetic linguistics (Köhler,

```
<!-- ... --> <DIV id="DIV1">
    <P id="P1">
        <S id="S1">
            <T c="l">
                <O>Montag</O><L p="Weekday">Montag</L>
            </T>
            <PUN>,</PUN><E/>
            <N>3</N>
            <PUN>.</PUN><E/>
            <T c="l">
                <O>Januar</O><L p="Month">Januar</L>
            </T><E/>
            <N>1994</N><E/>
            <T c="l">
                <O>NACHRICHTEN</O>
                <L p="N">Nachricht</L>
            </T><E/>
            <T c="l"><O>S\"{u}ddeutsche</O><L p="A">s\"{u}ddeutsch</L></T><E/>
            <T c="l"><O>Zeitung</O><L p="N">Zeitung</L></T>
        </S> <!-- ... -->
    </P> <!-- ... -->
</DIV> <!-- ... -->
```

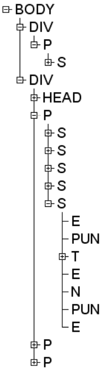

Figure 1: Outline of a sample text document (left) and its corresponding DOM-representation (right) – generated by the TextMiner system (Mehler, 2002) – in the form of an ordered rooted tree as input to feature selection (all XML element contents deleted).

1999) which develops reference systems of quantitative variables of dynamic linguistic systems.

2. *The tree kernel approach:* Secondly, we elaborate SVM learning which uses kernels that operate on pairs of examples (Vapnik, 1995). The theory of SVMs ensures that kernels can be defined for tree-like structures (Haussler, 1999; Jain et al., 2005; Schoelkopf and Smola, 2002). A large class of structure kernels is formed by convolution kernels (Haussler, 1999) including the one defined by Collins and Duffy (2001) for labeled, ordered trees. This tree kernel has previously been applied to structure based classification of sentences described by their parse trees (Collins and Duffy, 2001; Moschitti, 2004). In order to classify texts, we apply an extended version of this kernel in Section 3.2. This new kernel allows a variable number of descendants for some tree nodes.

Input to these two approaches are tree-like instances of the *Document Object Model* (DOM). That is, starting from a text corpus $C = \{x_1, \ldots, x_r\}$, each text $x_k \in C$ is mapped onto a DOM-tree representing its logical document structure. This is done by means of the TextMiner system (Mehler, 2002) which uses an element name-adapted version of XCES (Ide et al., 2000) in order to explore the paragraph structure of texts down to the level of their lexical tokens (by disregarding sentence structure). Figure (1) illustrates a sample text and its DOM-based representation in the form of an ordered rooted tree. Trees of this sort are subsequently input of feature selection, that is, of quantitative structure analysis (Section 3.1) and of tree kernel methods (Section 3.2).

3.1 Quantitative Structure Analysis

The *Vector Space Model* (VSM) is one of the most successful quantitative text representation models. It is the starting point of *Latent Semantic Analysis* (LSA) which takes an appropriately weighted term-document matrix as input and aims at eliminating as much of its noise as possible. From a linguistic point of view, the VSM performs a *bag-of-words* approach which focuses on *lexical* cohesion as a source of text similarity measuring (e.g., by the well-known cosine approach). LSA complements an effort in exploring indirect relations, e.g., of texts which may be said to be similar not because of having the same, but because of sharing similar lexical items whose similarity is computed in terms of their co-occurrence patterns. In this section, we present a likewise quantitative text representation model in terms of a *bag-of-structural-features approach* henceforth called *Quantitative Structure Analysis* (QSA). QSA is no longer based on lexical, but on structural features of the input texts. Thus, although we map texts onto feature vectors, their coefficients represent quantitative characteristics of text structure. Needless to say that both the VSM or LSA and the QSA approach can be combined (Mehler, 2002). However, in this paper we will concentrate on the separability potential of structural text features.

Generally speaking, QSA is based on a set $T_S = \{T_1, \ldots, T_m\}$ of structure types (e.g. constituency types) of some level S of text structuring (e.g. of the level of logical document structure or of intentional structure). For some structure type (e.g. sentence, paragraph, phrase) $T_i \in T_S$ of the level S we can ask, among other things, for (i) the *frequency*, (ii) (absolute) *length* (in terms of the number of leaf nodes), (iii) complexity (in terms of the number of immediate daughter nodes or some other mediate level if existing), (iv) *depth*, (v) *extension* (i.e. relative length), (vi) *proportion* (of text formation), (vii) *text position*, (viii) *distance*, (ix) (e.g. Markov) *order* or *arrangement* or (x) for the *characteristic repetition* (e.g. positional repetition) of instances of this type within a given text x of the corpus $C = \{x_1, \ldots, x_r\}$ and, thus, for different quantitative features. More specifically, if $T_i \in T_S$ is a structure type of level S and F_i is one of the latter features, we need to specify a measuring unit in order to calculate its value for a given instance of T_i in x. Let, for example, T_i be the structure type named title as part of the *Logical Document Structure* (LDS), then we can ask for its length in terms of lexical tokens or its frequency in terms of its number of occurrences. Analogously, we may ask for a title's depth in terms of the number of subtitles it is dominating. Alternatively, we may calculate its depth as the depth of its phrase structure tree. Another example is rhetorical structure in the sense of *rhetorical structure theory* (Mann and Thompson, 1988). In this case, we may ask for the text position of contrast relations. Now, the measuring unit is less clear so that we may define, for example, that the position of a contrast relation in a text equals the number of elementary text spans to its left. Following this procedure, we get a different positioning number for each title of the input text. In order to keep the presentation of our algorithm abstract, we resist defining the space of all possible measuring units for each of the features, but will define them as soon as needed.

The idea behind QSA is to, firstly, collect all values of a given structure feature in a text x where these values are, secondly, input to some aggregation functions in order to, thirdly, derive x's *Quantitative Structure Profile* (QSP). As we have to put apart the choice of text structure types and their features, we come up with a quadripartite approach:

Segmentation Let S be a description level of text structure and $T_S = \{T_1, \ldots, T_m\}$ a set of structure types of S. For each structure type $T_i \in T_S$ and each text $x_k \in C = \{x_1, \ldots, x_r\}$ we build a separate vector of instances of T_i in x_k. Using a functional notation, we write:

$$T_i(x_k) = \left(I^1_{x_k}, \ldots, I^h_{x_k}\right)' \tag{1}$$

where $I^l_{x_k}, l \in \{1, \ldots, h\}$, is the lth instance of T_i in x_k. We assume that the linear order $(1, \ldots, h)$ is defined by the order of occurrences of T_i's instances in x_k. Note that different texts may differ in the number of their instances of T_i. Now, let

$$\mathbb{T}_i(C) = \bigcup_{k=1}^{|C|} \{T_i(x_k)\} \tag{2}$$

Thus, we can write

$$T_i \colon C \rightarrow \mathbb{T}_i(C) \tag{3}$$

Feature Validation Now, let $F = \{F_1, \ldots, F_n\}$ be a set of numerical features (e.g. length, depth, complexity as enumerated above) and $F_j \in F$. Using once more a functional notation, we define

$$F_j \colon \mathbb{T}_i(C) \rightarrow \cup_{h=1}^{\infty} \mathbb{R}^h \tag{4}$$

by setting

$$F_j((I^1_{x_k}, \ldots, I^h_{x_k})') = (F_j(I^1_{x_k}), \ldots, F_j(I^h_{x_k}))' = \vec{v}(T_i, F_j, x_k) \in \mathbb{R}^h \tag{5}$$

$F_j(I^l_{x_k})$, $l \in \{1, \ldots, h\}$, is the F_j-value of the lth instance of T_i in x_k. So far, each text $x_k \in C$ is mapped for each feature $F_j \in F$ onto a separate vector of the F_j-values of T_i's instances in x_k. As these vectors may differ with respect to the number of their coefficients, we do not yet get a matrix.[2] The next step is to aggregate each of these vectors separately to get a single value for feature F_j of all instances of T_i in x_k. Thus, we conceive the vectors $F_j((I^1_{x_k}, \ldots, I^h_{x_k})') = \vec{v}(T_i, F_j, x_k) = \vec{v}_{ijk}$ as value distributions. In order to make these distributions comparable, we perform standardization by means of z-scores so

[2] An alternative would be to operate with empty coefficients – we do not follow this approach.

that random variables are derived with means of 0 and variances of 1. Without any loss of generality we assume henceforth that all vector coefficients are standardized and write

$$\mathbb{F}_j(\mathbb{T}_i(C)) = \bigcup_{k=1}^{|C|} \{\vec{v}_{ijk}\}$$

Feature Aggregation Now, let $O = \{O_1, \ldots, O_o\}$ be a set of parameters of location or statistical spread and $O_p \in O$ one of these aggregation functions. Then we define

$$O_p \colon \mathbb{F}_j(\mathbb{T}_i(C)) \rightarrow \mathbb{R} \tag{6}$$

where $O_p(\vec{v}_{ijk}) \in \mathbb{R}$ is the value of O_p when performed on the vector of the values of feature F_j of T_i's instances in x_k. So far, we mapped each of the features F_j onto a single number $O_p(\vec{v}_{ijk}) \in \mathbb{R}$. The final stage is to collect these numbers to get a quantitative structure profile for each text.

Text Representation For each text $x_k \in C$, we define a quantitative structure profile as

$$\begin{aligned} \mathrm{qsp}(x_k) \quad = \langle \quad & O_1(\vec{v}_{11k}), \ldots, O_1(\vec{v}_{1nk}), \ldots, O_1(\vec{v}_{m1k}), \ldots, O_1(\vec{v}_{mnk}), \\ & \ldots, \\ & O_o(\vec{v}_{11k}), \ldots, O_o(\vec{v}_{1nk}), \ldots, O_o(\vec{v}_{m1k}), \ldots, O_o(\vec{v}_{mnk}) \quad \rangle \\ \in \quad & \mathbb{R}^{m \cdot n \cdot o} \end{aligned} \tag{7}$$

That is, qsp is a function $\mathrm{qsp} \colon C \rightarrow \mathbb{R}^{m \cdot n \cdot o}$ which allows to build a $(|C|, m \cdot n \cdot o)$-matrix $\mathrm{qsp}(C) = (a_{ij})$ where a_{ij}, $j = (s-1)mn + (t-1)n + v$, is the value of aggregation function $O_s \in O$ performed on the feature distribution induced by the tth feature $F_t \in F$ with respect to instances of the vth text structure type $T_v \in T_S$ in text x_i. We call $\mathrm{qsp}(C)$ *the quantitative structure profile* of corpus C.

Note that matrix $\mathrm{qsp}(C)$ can be input to single value decomposition with subsequent noise reduction so that QSA is complemented by a latent variable analysis.

So far, we described a bag-of-features approach as input to supervised text categorization or unsupervised classification. In the following section, we describe alternative approaches to building kernels for mapping tree-like structures. In Section 4, these two approaches are evaluated.

3.2 Tree Kernels for XML Documents

In order to investigate the structure-based classification of XML documents based on their DOM trees (Document Object Model), we might apply the SVM (Vapnik, 1995) after defining an appropriate *tree kernel*. This can either be accomplished by directly defining an appropriate function that is positive-semidefinite (PSD, e.g., Schoelkopf and Smola 2002) or by explicitly defining an appropriate feature mapping for the structures considered, e.g., by means of patterns. An example of a kernel is the *parse tree kernel*

(Collins and Duffy, 2001; Moschitti, 2004), which is applicable to parse trees of sentences with respect to a given grammar.

In contrast to parse trees in which a grammar rule applied to a non-terminal determines number, type and sequence of the children, structural parts of a text represented by its DOM tree might have been deleted, permuted or inserted compared to a text considered similar. This higher flexibility should be taken into account in the similarity measure represented by the tree kernel, because otherwise the value for similar documents might be unreasonably small. Moreover, we might want to include textual information present in nodes by plugging in suitable kernels operating, e.g., on the usual TFIDF representation of the respective text, or more elaborate ones like string kernels (Lodhi et al., 2002) operating on the word sequence or even additional tree kernels operating on the parsed sentence structure.

We therefore extended previous work on tree kernels suitable for XML data in several respects that are useful in the context of HTML and XML documents. The *DOM tree kernel* (DomTK) is a straightforward generalization of the parse-tree kernel to DOM trees. The *set tree kernel* (SetTK) allows permutations of child subtrees in order to model document similarity more appropriately, but can still be computed relatively efficiently.

3.2.1 The Parse Tree Kernel

In the following, we consider trees whose nodes $v \in V$ are labeled by a function $\alpha : V \longrightarrow \Sigma$, where Σ is a set of node labels. The elements of Σ can be thought of as tuples describing the XML tag and attributes of a non-leaf node in the DOM tree. Leaves are usually labeled with words or parts of texts. We will incorporate node information by using a kernel k^{Σ} operating on pairs of node labels, i.e., on tags, attributes, and/or texts. Two trees T and T' are called isomorphic if there is a bijective mapping of the nodes that respects the structure of the edges, the labellings specified by α and α', and the ordering of the nodes.

Collins and Duffy (2001, 2002) defined a tree kernel that can be applied in the case of parse trees of natural language sentences (see also Moschitti 2004), in which non-leaf nodes are labeled with the non-terminal of the node, and leaves with words. The production applied to a non-leaf node determines the number, type, and ordering of the child nodes.

Collins and Duffy showed that $k(T, T')$ can be computed efficiently by determining the number of possible *mappings* of isomorphic partial parse trees (excluding such consisting of a single node only). Partial parse trees correspond to incomplete parse trees, in which leaves might be labeled with non-terminals. Let $v \in V$ and $v' \in V'$. The function $\Delta(v, v')$ is defined as the number of isomorphic mappings of partial parse trees rooted in v and v', respectively. Collins and Duffy stated in their article the fact that

Δ is a so-called convolution kernel (Haussler, 1999) having form

$$k(T,T') = \sum_{v \in V, v' \in V'} \Delta(v,v')\,. \tag{8}$$

The Δ-function can be computed recursively by setting $\Delta(v,v') = 0$ for *any* words and if the productions applied in v and v' are different. Different productions mean different non-terminals, or identical non-terminals but different grammar rules (i.e., the number or type of corresponding child nodes do not correspond). If the productions in v and v' are identical and both nodes are pre-terminals, we set $\Delta(v,v') = 1$. For non-terminals with identical productions, Collins and Duffy use the recursive definition

$$\Delta(v,v') = \prod_{i=1}^{n(v)} (1 + \Delta(v_i, v'_i))\,, \tag{9}$$

where v_i is the i-th child of v, and v'_i is the i-th child of v'. $n(v)$ denotes the number of children of v (corresponding to that of v').

It is possible to down-weight deeper trees using a factor $\lambda \in [0,1]$. The corresponding recursive computation is $\Delta(v,v') = \lambda \prod_{i=1}^{n(v)}(1 + \Delta(v_i, v'_i))$ together with the modified base case $\Delta(v,v') = \lambda$ for pre-terminals with identical productions.

3.2.2 Kernels for DOM trees

The *DOM tree kernel* (DomTK) is a relatively straightforward extension of the parse tree kernel that allows to incorporate node labels by means of k^{Σ}. This achieved by defining $\Delta_{\mathrm{DomTK}}(v,v') = \lambda \cdot k^{\Sigma}(\alpha(v), \alpha(v'))$ for nodes. If not both v and v' are leaves, we, in contrast to the parse tree kernel, compare just as many children as possible using the given order $\leq$ on the child nodes.

It can be seen from a corresponding feature mapping that when comparing two trees T and T', we have two take into account shorter prefixes of the child tree sequences of two nodes v and v' as well. This is done by defining the Δ-function as

$$\Delta_{\mathrm{DomTK}}(v,v') = \lambda \cdot k^{\Sigma}(\alpha(v), \alpha(v'))\Big(1 + \sum_{k=1}^{\min(n(v),n'(v'))} \prod_{i=1}^{k} \Delta_{\mathrm{DomTK}}(v_i, v'_i)\Big) \tag{10}$$

for all nodes v and v'.

The DOM tree kernel does not allow the child trees of v and v' to be permuted without a high loss in similarity as measured by the kernel value $k(T,T')$. This behavior can be improved, however, by considering the child tree sequences as *sets* and applying a so-called set kernel to them, which is also an instance of the convolution kernel. This

mean number of articles per rubric	1426.8
standard deviation	2315.9
$\mu - \frac{\sigma}{2}$	268.8303
$\mu + \frac{\sigma}{2}$	2584.8

Table 1: Values of the parameters of the procedure of category selection.

results in the definition

$$\Delta_{\text{SetTK}}(v, v') = \sum_{i=1}^{n(v)} \sum_{i'=1}^{n'(v')} \Delta_{\text{SetTK}}(v_i, v'_{i'}), \tag{11}$$

i.e., all possible pairwise combinations of child trees are considered.

When looking for a suitable feature space in the case $\lambda = 1$ and $k^{\Sigma} = k^{id}$ where $k^{id} = 1$ for nodes with identical labels, and $k^{id} = 0$ otherwise, we find that the definition in (11) corresponds to considering *paths* from the root to the leaves. This is a well-known technique for characterizing labeled graphs (see, e.g., Geibel and Wysotzki 1996), which can also be applied to trees. We also investigated tree kernels based on string kernels as in Kashima and Koyanagi (2002) and Moschitti (2006), but found them too inefficient for the application at hand.

4 Evaluation

The main hypothesis of our approach is that structure-based classification is a serious alternative to the bag-of-lexical-features approach. Thus, we expect a high selectivity of structural features with respect to functionally delimitable text types. In order to support this hypothesis, we process a corpus of press communication. The corpus is built as follows: We start from a ten years release of the German newspaper Süddeutsche Zeitung (SZ) and select *all* articles of *all* rubrics within this corpus. This gives a corpus of $135,546$ texts of 96 rubrics. Note that each text is mapped onto exactly one rubric. As the frequency distribution of the rubrics is unbalanced (it ranges from a rubric with only 2 instances to rubrics with more than 10,000 instances) and since the number of categories is – compared to other experiments in the field of text classification – large, we decided to select a subset of rubrics as target categories to be learnt. This was done as follows: We computed the mean μ and standard deviation σ of each rubric in terms of the number of its text instances and chose those rubrics R whose cardinality $|R|$ behaves as follows (cf. Table 1):

$$\mu - \sigma/2 < |R| < \mu + \sigma/2$$

As a result, we select 31 rubrics as target categories (cf. Table 2). This generates a corpus C of $31,250$ texts.

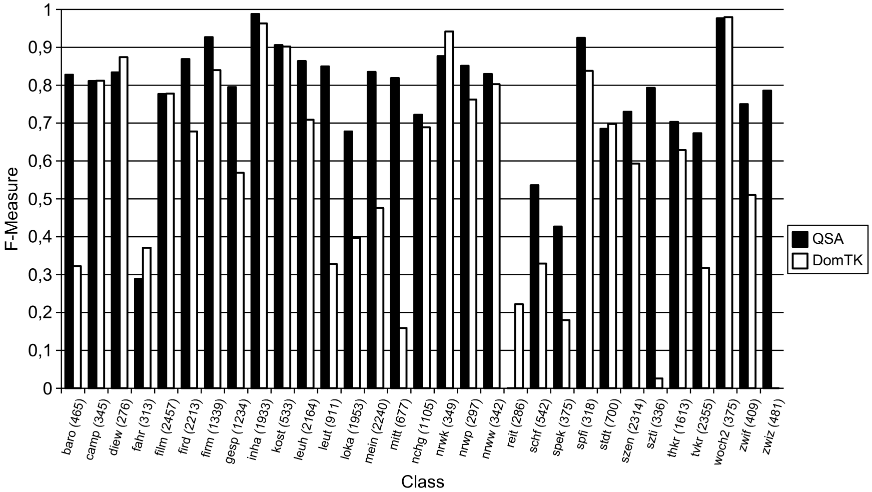

Table 2: Results of two categorization experiments using 31 rubrics of the SZ: *QSA* (black bars) and *DomTK* (gray bars). Bars are ordered alphabetically according to the code name of the category. Numbers in parentheses denote the size (number of instances) of the corresponding category.

Next, we perform an SVM-experiment in the framework of QSA. That is, we use $\mathrm{qsp}(x)$ as the vector representation of texts $x \in C$. More specifically, we refer to logical document structure (LDS) as the focal level of text structuring and set $T_{LDS} = \{$division, paragraph, sentence, headline sentence, headline paragraph$\}$. Next, we set $F = \{$complexity, length$\}$ and $O = \{$mean, standard deviation, entropy$\}$. Thus, for each input text we get a vector $\mathrm{qsp}(x)$ with exactly 30 features.

The design parameters of the subsequent SVM-experiment are as follows: We use an RBF-kernel with $\gamma = 0.00001$ and a trade-off between training error $c = 1000$. Further, we train a binary classifier for each of the 31 categories. Thus, for input corpus C and any rubric c_i of the set of target categories $\mathbb{C}$ the set of negative examples of c_i is set to $C \setminus [c_i]$ where $[c_i] \subseteq C$ is the set of all instances of c_i in C. In the present experiment, all sets $[c_i]$ are pairwise disjunct. Next, we utilize the *leave-on-out cross-validation* method (Hastie et al., 2001) and get a recall and precision value for each of the categories trained by means of the SVMlight (Joachims, 2002). This allows us to compute an F-score (FS) for each of the rubrics $c_i \in \mathbb{C}$ separately as (Hotho et al., 2005):

$$FS_i = \frac{2}{\frac{1}{\mathrm{recall}_i} + \frac{1}{\mathrm{precision}_i}}$$

The F-scores of the 31 categories are summarized in Table 2. Finally, we set $\mathbb{L} = \{[c_i] \,|\, c_i \in \mathbb{C}\}$ – obviously, $\mathbb{L}$ is a partition of C – and compute the F-Measure as a weighted mean of the F-scores of all categories as:

$$\text{F-Measure}(\mathbb{L}) = \sum_{i=1}^{|\mathbb{C}|} \frac{|[c_i]|}{|C|} FS_i$$

In the framework of QSA, this gives an overall F-measure of SVM-based learning of 0.78 – a remarkably good results for a rather large set of different categories to be learnt. Note that we took all 31,250 texts of the corpus into account in order to compute this result.

Next, we perform a comparable experiment using the tree kernels as introduced in Section 3.2. The complexity of computing $k(T, T')$ based on (8) depends on the product of the node numbers $n(v)n(v')$, see Collins and Duffy (2001). Since some of the trees in the corpus are relatively large, we had to down-sample the corpus to a subset containing only 6250 examples. Instead of using leave-one-out cross-validation, we used 10-fold cross validation, which is more efficient, but known to produce reliable results, too. In order to make up for this loss in data to some extend, we performed a coarse search for optimal values of λ (parameter for tree depth, see above) and C. We varied λ additively in the interval $[0.0, 2.0]$ and C multiplicatively in the interval $[0.0001; 400.0]$. For SetTK we only used a subset of the parameters, because its complexity depends quadratically on the branching factor. For DomTK, choosing λ around 0.5 and C around 50 produced reasonable results for many classes. In addition to DomTK and SetTK, we also tested an implementation of a tree kernel based on a string kernel (cf. Moschitti 2006). The computation of a single kernel matrix took more than three days, so we are not able to present results for this third kernel in this article. Notice that for the second part of our SVM experiment based on tree kernels we used the LIBSVM (Chang and Lin, 2001).

4.1 Discussion

Table 2 presents the results of the experiment for QSA and the DomTK approach. Every category is identified by a short-cut representing a rubric (e.g., `woch2` = '*Wochenchronik*' – '*chronicle of the week*'). The corresponding F-Score values demonstrate the separability of most of the categories. Although DomTK performs better for a few classes, it is usually outperformed by QSA. This confirms results also found in other areas where tree kernel methods often perform worse than feature-based methods. Note that DomTK had to operate on a down-sampled data set (and fewer parameter combinations could be tried, too), while QSA explored the whole spectrum of the input corpus.

In the case of QSA, half of the categories perform with an F-score above 0.8 (vs. nine in the case of DomTK) – five (four in the case of DomTK) categories lead to F-score values above 0.9. The combined F-measure value of the QSA approach is 0.78. This shows that there are many categories which can be reliably attributed to their category by only looking for a small set of their quantitative structural features. Obviously, the set of 30 features taken into account by the present instance of QSA is much smaller than by VSM which may take several thousand lexical dimensions into account. Results for other text types (genres and registers) give a comparable result so that the method is, obviously, not restricted to the area of press communication (Pustylnikov and Mehler, 2007). However, in the case of the DomTK and the QSA approach there are poorly performing categories. This is *not* surprising as we do not expect that structure is the only reliable manifestation of text types. Rather we shed light on its potential which in future work will be combined with content-related approaches to text classification.

5 Conclusion

This paper evaluated logical document structure as a source of feature selection in text classification. It has shown that structure-based classifications come into reach and produce very promising results. This finding is all the more important as, e.g., QSA provides an easy to compute and space efficient text representation model. Thus, the paper is a first step towards the far-reaching goal of developing a prototype ontology of text types. Future work will focus on elaborating the present approach, especially in terms of a sensitivity analysis of the whole spectrum of quantitative text characteristics. Further, we will develop a corresponding graph model of web documents.

Acknowledgments

We thank Alessandro Moschitti (University of Rome) for helpful discussions. We thank Sebastian Herold (University of Osnabrück) for performing the kernel experiments. AM and OP's work is supported by the Research Group 437 *Text Technological Modeling of Information* (`http://www.text-technology.de/`) and by the SFB 673 *Alignment in Communication* (`http://ariadne.coli.uni-bielefeld.de/sfb/`) both funded by the German Research Foundation (DFG).

References

Baayen, H., van Halteren, H., and Tweedie, F. (1996). Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution. *Literary and Linguistic Computing*, 11(3):121–131.

Baeza-Yates, R. and Ribeiro-Neto, B., editors (1999). *Modern Information Retrieval*. Addison-Wesley, Reading, Massachusetts.

Bateman, J. A., Kamps, T., Kleinz, J., and Reichenberger, K. (2001). Towards constructive text, diagram, and layout generation for information presentation. *Computational Linguistics*, 27(3):409–449.

Biber, D. (1995). *Dimensions of Register Variation: A Cross-Linguistic Comparison.* Cambridge University Press, Cambridge.

Brinker, K. (1992). *Linguistische Textanalyse. Eine Einführung in Grundbegriffe und Methoden.* Erich Schmidt, Berlin.

Chang, C.-C. and Lin, C.-J. (2001). *LIBSVM: a library for support vector machines.* Software available at`http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Collins, M. and Duffy, N. (2001). Convolution kernels for natural language. In *NIPS*, pages 625–632.

Collins, M. and Duffy, N. (2002). New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *ACL*, pages 263–270.

Dehmer, M. (2005). *Strukturelle Analyse Web-basierter Dokumente.* Multimedia und Telekooperation. DUV, Berlin.

Geibel, P. and Wysotzki, F. (1996). Learning relational concepts with decision trees. In Saitta, L., editor, *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 166–174, San Fransisco, CA. Morgan Kaufmann Publishers.

Gleim, R., Mehler, A., and Dehmer, M. (2006). Web corpus mining by instance of wikipedia. In Kilgariff, A. and Baroni, M., editors, *Proceedings of the EACL 2006 Workshop on Web as Corpus, April 3-7, 2006, Trento, Italy*, pages 67–74.

Halliday, M. A. K. and Hasan, R. (1989). *Language, Context, and Text: Aspects of Language in a Socialsemiotic Perspective.* Oxford University Press, Oxford.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The elements of statistical learning. Data Mining, Inference, and Prediction.* Springer, Berlin/New York.

Haussler, D. (1999). Convolution kernels on discrete structures. Technical Report UCS-CRL-99-10, UC Santa Cruz.

Heinemann, W. (2000). Textsorte – Textmuster – Texttyp. In Brinker, K., Antos, G., Heinemann, W., and Sager, S. F., editors, *Text- und Gesprächslinguistik. Linguistics of Text and Conversation*, pages 507–523. De Gruyter, Berlin/New York.

Hotho, A., Nürnberger, A., and Paaß, G. (2005). A Brief Survey of Text Mining. *LDV-Forum*, 20(1):19–62.

Ide, N., Bonhomme, P., and Romary, L. (2000). XCES: An XML-based standard for linguistic corpora. In *Proc. of LREC 2000, Athens*, pages 825–830.

Jain, B. J., Geibel, P., and Wysotzki, F. (2005). SVM learning with the Schur-Hadamard inner product for graphs. *Neurocomputing*, 64:93–105.

Joachims, T. (2002). *Learning to classify text using support vector machines.* Kluwer, Boston.

Karlgren, J. (1999). Non-topical factors in information access. In *WebNet (1)*, pages 27–31.

Kashima, H. and Koyanagi, T. (2002). Kernels for semi-structured data. In *ICML*, pages 291–298.

Kessler, B., Nunberg, G., and Schütze, H. (1997). Automatic detection of text genre. In *Proceedings of the 35th ACL and 8th EACL, Madrid*, pages 32–38.

Köhler, R. (1999). Syntactic structures. properties and interrelations. *Journal of Quantitative Linguistics*, 6:46–57.

Lee, Y.-B. and Myaeng, S. H. (2002). Text genre classification with genre-revealing and subject-revealing features. In *Proc. of the 25th Annual International ACM SIGIR Conf. on Research and Development in IR*, pages 145–150. ACM Press.

Lewis, D. D. (1992). Feature selection and feature extraction for text categorization. In *Proceedings of Speech and Natural Language Workshop*, pages 212–217.

Lindemann, C. and Littig, L. (2006). Coarse-grained classification of web sites by their structural properties. In *Proc. of WIDM'06*, pages 35–42.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. J. C. H. (2002). Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444.

Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8:243–281.

Martin, J. R. (1992). *English Text. System and Structure.* John Benjamins, Philadelphia.

Mehler, A. (2002). Hierarchical orderings of textual units. In *Proc. of COLING'02*, pages 646–652.

Mehler, A. (2007). Structure formation in the web. In Witt, A. and Metzing, D., editors, *Linguistic Modeling of Information and Markup Languages.* Springer, Dordrecht.

Mehler, A., Gleim, R., and Dehmer, M. (2006). Towards structure-sensitive hypertext categorization. In *Proc. of the 29th Annual Conf. of the GfKl, March 9-11, 2005, Universität Magdeburg*, pages 406–413.

Miller, G. A. and Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

Moschitti, A. (2004). A study on convolution kernels for shallow semantic parsing. In *Proc. of the 42th Conf. of the ACL*, pages 335–342.

Moschitti, A. (2006). Efficient convolution kernels for dependency and constituent syntactic trees. In *ECML*, pages 318–329.

Power, R., Scott, D., and Bouayad-Agha, N. (2003). Document structure. *Computational Linguistics*, 29(2):211–260.

Pustylnikov, O. (2006). How much information is provided by text structure? Automatic text classification using structural features (in German). Master thesis, University of Bielefeld, Germany.

Pustylnikov, O. and Mehler, A. (2007). A new look on register analysis: text classification by means of structural classifiers. In *Proceedings of the 10th International Pragmatics Conference (Göteborg, 8-13 July 2007).*

Santini, M. (2007). Characterizing genres of web pages: Genre hybridism and individualization. In *Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS'07).*

Schoelkopf, B. and Smola, A. J. (2002). *Learning with Kernels.* The MIT Press, Cambridge, MA.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

Sowa, J. F. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations.* Brooks/Cole, Pacific Grove.

Tuldava, J. (1998). *Probleme und Methoden der quantitativ-systemischen Lexikologie.* Wissenschaftlicher Verlag, Trier.

van Dijk, T. A. and Kintsch, W. (1983). *Strategies of Discourse Comprehension.* Academic Press, New York.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory.* Springer, New York.

Ventola, E. (1987). *The Structure of Social Interaction: a Systemic Approach to the Semiotics of Service Encounters.* Pinter, London.

Wolters, M. and Kirsten, M. (1999). Exploring the use of linguistic features in domain and genre classification. In *Proceedings of the EACL*, pages 142–149.