Exploring the Use of Linguistic Features in Domain and Genre Classification

Maria Wolters[1] and **Mathias Kirsten**[2]
[1]Inst. f. Kommunikationsforschung u. Phonetik, Bonn; wolters@ikp.uni-bonn.de
[2]German Natl. Res. Center for IT-AiS.KD-, St. Augustin; mathias.kirsten@gmd.de

Abstract

The central questions are: How useful is information about part-of-speech frequency for text categorisation? Is it feasible to limit word features to content words for text classifications? This is examined for 5 domain and 4 genre classification tasks using LIMAS, the German equivalent of the Brown corpus. Because LIMAS is too heterogeneous, neither question can be answered reliably for any of the tasks. However, the results suggest that both questions have to be examined separately for each task at hand, because in some cases, the additional information can indeed improve performance.

1 Introduction

The greater the amounts of text people can access and have to process, the more important efficient methods for text categorisation become. So far, most research has concentrated on content-based categories. But determining the **genre** of a text can also be very important, for example when having to distinguish an EU press release on the introduction of the euro from a newspaper commentary on the same topic.

The results of e.g. (Lewis, 1992; Yang and Pedersen, 1997) indicate that for good content classification, we basically need a vector which contains the most relevant words of the text. Using n-grams hardly yields significant improvements, because the dimension of the document representation space increases exponentially. But do word-based vectors also work well for genre detection? Or do we need additional linguistically motivated features to capture the different styles of writing associated with different genres?

In this paper, we present a pilot study based on a set of easily computable linguistic features, namely the frequency of part-of-speech (POS) tags, and a corpus of German, LIMAS (Glas, 1975), which contains a wide range of different genres. LIMAS is described briefly in Sec. 3, while sections 2 and 4 motivate the choice of features. The text categorisation experiments are described in Sec. 5.

2 Linguistic Cues to Genre

2.1 What is genre?

The term "genre" is more frequent in philology and media studies than in mainstream linguistics (Swales, 1990, p.38). When it is not used synonymously with the terms "register" or "style", genre is defined on the basis of *non-linguistic* criteria. For example, (Biber, 1988) characterises genres in terms of author/speaker purpose, while text types classify texts on the basis of text-internal criteria. Swales phrases this more precisely: Genres are collections of communicative events with shared communicative purposes which can vary in their prototypicality. These communicative purposes are determined by the discourse community which produces and reads texts belonging to a genre.

But how can we extract its communicative purpose from a given text? First of all, we need to define the genres we want to detect. The definitions which were used in this study are summarised in section 3.1. If we assume that the culture-specific conventions which form the basis for assigning a given text to a certain genre are reflected in the style of the text, and if that style can be characterised quantitatively as a tendency to favour certain linguistic options over others (Herdan, 1960), we can then proceed to search for linguistic features which both discriminate well between our genres and can also be computed reliably from unannotated text. Potential sources for such options are comparative genre studies (Biber, 1988), authorship attribution research (Holmes, 1998; Forsyth and Holmes, 1996), content analysis (Martindale and MacKenzie, 1995), and quantitative stylistics (Pieper, 1979). For the last step, classification, we need a robust statistical method which should preferably work well on sparse and noisy data. This aspect will be discussed in more detail in section 5.

In their paper on genre categorization, (Kessler et al., 1997) take a somewhat different approach. They classify texts according to generic facets. Those facets express distinctions that "answer to certain practical interests" (p. 33). The "brow" facet roughly corresponds to register, and the "narrative" facet is taken from text type theory, while the "genre" facet most closely correspond to our usage of the term.

2.2 Choice of features

There are two basic types of features: ratios and frequencies. Typical ratios are the type/token ratio, sentence length (in words per sentence), or word length (in characters per words). More elaborate ratios which have been found to be useful in quantitative stylistics (Ross and Hunter, 1994) are e.g. the ratio of determiners to nouns or that of auxiliaries to VP heads.

The most common features to be counted are words, or, more precisely, word stems. While most text categorisation research focusses on content words, function words have proved valuable in authorship attribution. The rationale behind this is that authors monitor their use of the most frequent words less carefully than that of other words. But this is not the reason why function words might prove to be useful in genre analysis. Rather, they indicate dimensions such as personal involvement (heavy use of first and second person pronouns), or argumentativity (high frequency of specific conjunctions). Content analysis counts the frequency of words which belong to certain diagnostic classes, such as for example aggressivity markers. The frequency of other linguistic features such as part-of-speech (POS), noun phrases, or infinitive clauses, has been examined selectively in quantitative stylistics. In his comparative analysis of written and spoken genres in English, Biber (Biber, 1988) lists an impressive array of 67 linguistically motivated features which can be extracted reliably from text. However, he sometimes relies heavily on the fixed word order of English for their computation, which makes them difficult to transfer to a language with a more flexible word order, such as German. (Karlgren and Cutting, 1994) reports good results in a genre classification task based on a subset of these features, while (Kessler et al., 1997) show that a prudent selection of cues based on words, characters, and ratios can perform at least equally well.

In our paper, we explore a hybrid approach. Starting from the classical information retrieval representation of texts as vectors of word frequencies (Salton and McGill, 1983), we explore how performance is affected if we include

- function word frequencies. For example, texts which aim at generalisable statements may contain more indefinite articles and pronouns and less definite articles.
- POS frequencies. (This essentially condenses information implicitly available in the word vector.) For example, nominal style should lead to a higher frequency of nouns, whereas descriptive texts may show more adjectives and adverbials than others.

Note that we do not experiment with sophisticated feature selection strategies, which might be worthwhile for the POS information (cf. Sec. 4). POS frequency information is the only higher–level linguistic information which is encoded explicitly. Most current POS–taggers are reliable enough (at least for English) for their output to be used as the basis for a classification, whereas robust, reliable parsers are hard to find. Another source of information would have been the position of a word in a sentence, but incorporating this would have lead to substantially larger feature spaces and will be left to future work. Semantic classes were not examined, because defining, building, fine-tuning, and maintaining such word lists can be an arduous task (cf. e.g. (Klavans and Kan, 1998)), which should therefore only be undertaken for corpora with both well-defined and well-represented genres, where inherently fuzzy class boundaries are less likely to counteract the effect of careful feature selection.

3 The LIMAS corpus of German

Since our focus is on genre detection, we decided not to use common benchmark collections such as Reuters[1] and OHSUMED[2] because they are rather homogenous with respect to genre.

LIMAS is a comprehensive corpus of contemporary written German, modelled on the Brown corpus (Kučera and Francis, 1967) and collected in the early 1970s. It consists of 500 sources with around 2000 words each. It has been completely tagged with POS tags using the MALAGA system (Beutel, 1998). MALAGA is based on the

[1] http://www.research.att.com/lewis/reuters21578.html

[2] ftp://medir.ohsu.edu/pub/ohsumed

STTS tagset for German which consists of 54 categories (Schiller et al., 1995). The corpus has already been used for text classification by (von der Grün, 1999).

Since the corpus is rather heterogeneous, we defined two sets of tasks, one based on the full corpus (CL), the other based on all texts from the categories law, politics, and economy (LPE) (104 sources in all). In the LPE experiments, emphasis was on searching for good parameters for the various learning algorithms as well as on the contribution of POS and punctuation information to classification accuracy. The experiments on the complete corpus, on the other hand, focus more on composition of the feature vectors.

3.1 Genre Classes

LIMAS is based on the 33 main categories of the Deutsche Bibliographie (German bibliography). Each of the bibliography's categories is represented according to its frequency in the texts published in 1970/1971, so that the corpus can be considered representative of the written German of that time (Bergenholtz and Mugdan, 1989). Furthermore, the corpus designers took care to cover a wide range of genres within each subcategory. As a result, groups of more than 10 documents taken from LIMAS will be rather heterogeneous. For example, press reports can be taken from broadsheets or tabloids, they can be commentaries, news reports, or reviews of cultural events.

Many of the main categories correspond to domains such as "mathematics" or "history". Although not evident from the category label, genre distinctions can also be quite important for domain classification, because some domains have developed specific genres for communication within the associated community. There are three such *domain* categories in our experiments, politics (P), law (L), and economy (E). Two further categories are academic texts from the humanities (H) and from the field of science and technology (S). In the LPE corpus, this distinction is collapsed into "academic" (A), the set of all scholarly texts in the corpus. Four categories are based on *genre* only. On one hand, we have press texts (N), and more specifically NH, press texts from high quality broadsheets and magazines, on the other hand, fiction (F) and FL, a low-quality subset of F. For LPE, we defined a category D consisting of articles from quality broadsheets. Table 1 gives an overview of the categories and the number of documents in each category for each corpus. In all subsequent experiments, we assume as baseline the classification accuracy which we get when all documents are assigned to the majority class. The baselines are specified in Tab. 1.

	L	P	E	H	S
CL n	20	44	40	109	72
CL acc.	96	91,2	92	78	85,6
	F	FL	N	NH	
CL n		60	26	53	30
CL acc.		88	94,8	89,4	94
	L	P	E	A	D
LPE n	20	43	40	45	26
LPE acc.	80	58,7	61,5	56,7	75

Table 1: Number of documents *n* in each category and classification accuracy *acc.* if each document is judged *not* to belong to that category.

4 Validating the Features

If the frequency of POS features does not vary significantly between categories, adding such information increases both random variation in the data as well as its dimensionality. To check for this, we conducted a series of non-parametric tests on CL for each POS tag.

In addition, binary classification trees were grown on the complete set of documents for each category, and the structure of the tree was subsequently examined. Classification trees basically represent an ordered series of tests. Each tree node corresponds to one test, and the order of the tests is specified by the tree's branches. All tests are binary. The outcome of a test higher up in the tree determines which test to perform next. A data item which reaches a leaf is assigned the class of the majority of the items which reached it during training. The trees were grown using recursive partitioning; the splitting criterion was reduction in deviance. Using the Gini index led to larger trees and higher misclassification rates. Since the primary purpose of the trees was not prediction of unseen, but analysis of seen data, they were not pruned. There were no separate test sets.

We tested for 12 categories and all STTS POS tags if the distribution of a tag significantly differs between documents in a given category and documents not in that category. These categories consist of the nine defined in Sec. 3 plus the content-based domains (Hi) and religion (R), and texts from tabloids and similar publications (PL).

Choice of Feature Values: The value of a feature is its relative frequency in a given text. The frequencies were standardised using z-scores, so that the resulting random variables have a mean of 0 and a variance of 1. The z-scores were rounded down to the next integer, so that all features whose frequency does not deviate greatly from the mean have a value of 0. Z-scores were computed on the basis of all documents to be compared. This makes sense if we view style as deviation from a default, and such defaults should be computed relative to the complete corpus of documents used, not relative to specific classification tasks.

Results: In general, only 7 of all 54 tags show significant differences in distribution for more than half of the categories, and the actual differences are far smaller than a standard deviation. However, for most tasks, there are at least 15 POS tags with characteristic distributions, so that including POS frequency information might well be beneficial.

The four most important content word classes are VVFIN (finite forms of full verbs), NN (nouns), ADJD (adverbial adjectives), and ADJA (attributive adjectives). Importance is measured by the number of significant differences in distribution. A higher incidence of VVFIN characterises F, FL, and NL, whereas texts from academia or about politics and law show significantly less VVFIN. The difference between the means is around 0.2 for F and FL, and below 0.1 for the rest. (Numbers relate to the z-scores). Note that we cannot claim that more VVFIN means less nouns (NN): scholarly texts both show less VVFIN and less NN than the rest of the corpus. For adjectives, we find that academic texts are significantly richer in ADJA (differences between 0.02–0.04), while FL contains more adverbial adjectives (difference 0.04).

But function words can be equally important indicators, especially personal pronouns, which are usually part of the stop word list. They are significantly less frequent in academic texts and categories E, L, NH, and P, and more frequent in fiction, NL, and R. Again, all differences are at or below 0.1. A lower frequency of personal pronouns can indicate both less interpersonal involvement and shorter reference chains.

Other valuable categories are, for example, pronominal adverbs (PAV) and infinitives of auxiliary verbs (VAINF), where the difference between the means usually lies between 0.2 and 0.4 for significant differences. (We restrict ourselves to discussing these in more detail for reasons of space.) Pronominal adverbs such as "deswegen" (because of this) are especially frequent in texts from law and science, both of which tend to contain texts of argumentative types. The frequency of infinitives of auxiliaries reflects both the use of passive voice, which is formed with the auxiliary "werden" in German, and the use of present perfect or pluperfect tense (auxiliary "haben"). In this corpus, texts from the domains of law and economy contain more VAINF than others.

The potential meaning of common punctuation marks is quite clear: the longer the sentences an author constructs, the fewer full stops and the more commata and subordinating conjunctions we find. However, the frequency of full stops is distinctive only for four categories: L, E, and H have significantly fewer full stops, NL has significantly more. We also find significantly more commata in fiction than in non-fiction, Possible sources for this are infinitive clauses and lists of adjectives.

With regard to the trees, we examined only those splits that actually discriminate well between positive and negative examples with less than 40% false positives or negatives. We will not present our analyses in detail, but illustrate the type of information provided by such trees with the category F. For this category, PPER, KOMMA, PTKZU ("to" before infinitive), PTKNEG (negation particle), and PWS (substituting interrogative pronoun) discriminate well in the tree. In the case of PTKZU and PTKNEG, this difference in distribution is *conditional*, it was not observed in the significance tests and surfaced only through the tree experiments.

5 Text Categorisation Experiments

For our categorisation experiments, we chose a relational k-nearest-neighbour (k-NN) classifier, RIBL (Emde and Wettschereck, 1996; Bohnebeck et al., 1998), and two feature-based k-NN algorithms, learning vector quantisation (LVQ, (Kohonen et al., 1996)), and IBL1(-IG) (Daelemans et al., 1997; Aha et al., 1991). The reason for choosing k-NN-based approaches is that this algorithm has been very successful in text categorisation (Yang, 1997).

We first ran the experiments on the LPE-corpus, which had mainly exploratory character, then on the complete corpus.

In the LPE-experiments, we distinguished six feature sets: CW, CWPOS, CWPP, WS, WS-POS, and WSPP, where CW stands for content word lemmata, WS for all lemmata, POS for POS information, and PP for POS and punctuation information.

In the CL-experiments, we did not control for the potential contribution of punctuation features to the results, but on the type of lemma from which the features were derived. We again explored 6 feature sets, CW, CWPOS, WS, WSPOS, FW, and FWPOS, where FW stands for function word lemmata. Punctuation was included in conditions WS, WSPOS, FW, and FWPOS, but not in CW and CWPOS. In addition to feature type, we also varied the length of the feature vectors.

In the following subsections, we outline our general method for feature selection and evaluation and give a brief description of the algorithms used. We then report on the results of the two suites of experiments.

5.1 Feature Selection

The set of all potential features is large - there are more than 29000 lemmata in the LPE corpus, and more than 80000 in the full corpus.

In a first step we excluded for the LPE corpus, all lemmata occuring less than 5 times in the texts, and for the CL corpus, all lemmata occurring in less than 10 sources, which left us with 4857 lemmata for LPE and 5440 lemmata and punctuation marks for CL. We then determined the relevance of each of these lemmata for a given classification task by their gain ratio (Yang and Pedersen, 1997). From this ranked list of lemmata, we constructed the final feature sets.

5.2 The Algorithms

RIBL: RIBL is a k-NN classification algorithm where each object is represented as a set of ground facts, which makes encoding highly structured data easier. The underlying first-order logic distance measure is described in (Emde and Wettschereck, 1996; Bohnebeck et al., 1998). Features were not weighted because using Kononenko's Relief feature weighting (Kononenko, 1994) did not significantly affect performance in preliminary experiments. The input for RIBL consists of three relations lemma(di,lemma,v), pos(di,POS-Tag,v), and document(di), with di the document index and v the standardised frequency, rounded to the next integer value. In the CL experiments, the lemma tag covers both real lemmata and punctuation marks, in LPE, punctuation marks had a separate precidate. Relations with a feature value of 0 are omitted, reducing the size of the input considerably. For these features, a true relational representation is not necessary, but that might change for more complex features such as syntactic relations.

IBL: IBL stores all training set vectors in an instance base. New feature vectors are assigned the class of the most similar instance. We use the Euclidean distance metric for determining nearest neighbours. All experiments were run with (IBL-IG) or without (IBL) weighting the contribution of each feature with its gain ratio.

LVQ: LVQ also classifies incoming data based on prototype vectors. However, the prototypes are not selected, but *interpolated* from the training data so as to maximise the accuracy of a nearest-neighbour classifier based on these vectors. During learning, the prototypes are shifted gradually towards members of the class they represent and away from members of different classes. There are three main variants of the algorihm, two of which only modify codebook vectors at the decision boundary between classes.

5.3 LPE-Experiments

5.3.1 Procedure

From the complete set of documents, we constructed three pairs of training and test sets for training the feature classifiers. The test sets are mutually disjunct; each of them contains 5 positive and 5 negative examples. The corresponding training sets contain the remaining 95 documents. For RIBL, test set performance is determined using leave-one-out cross validation. Feature vectors contained either 100, 500, or 1000 lemma features.

On the basis of test set performance, we determined precision, recall, and accuracy. Instead of determining recall/precision breakeven point as in (Joachims, 1998) or average precision over different recall values as in (Yang, 1997), we provide both values to determine which type of error an algorithm is more susceptible to. Tab. 2 summarizes the results.

5.3.2 Algorithm-specific results

Condition IBL-IG resulted in significantly higher precision (+0.5%) than IBL, but lower recall and accuracy (difference not significant). The number of neighbouring vectors was also varied ($k = 1, 3, 5, 7$). For precision, recall, and accuracy, best results were achieved with $k = 3$. A pure nearest-neighbour approach led to classifying all examples as negative. The number of neighbours k was also varied for RIBL. Contrary to IBL, it performs best for $k = 1$.

For the LVQ runs, we used the variant OLVQ1. In this algorithm, one codebook vector is adapted at a time; the rate of codebook vector adaptation is optimised for fast convergence. The resulting codebook was not tuned afterwards to avoid overfitting. We varied both the number of codebook vectors (10,20,50,90) and the initialisation procedure: during one set of runs, each class receives the same number of vectors, during the other, the number of codebook vectors is proportional to class size. Performance increases if codebook vec-

Task	Alg.	Prec.	Recall	FN	FS
A	RIBL	92,9	94,05	100	wspos
	IBL	75	75	1000	ws*
	LVQ	99,67	100	500	cwpos
E	RIBL	97,59	77,18	500	ws
	IBL	75	75	1000	all
	LVQ	100	100	1000	all
L	RIBL	95,45	100	100	wspos
	IBL	75	75	100/1000	all
	LVQ	100	100	100	ws*
N	RIBL	100	100	100	wspos
	IBL	75	75	100	all
	LVQ	100	100	100	all
P	RIBL	96,93	89,09	500	ws
	IBL	75	75	100/1000	all
	LVQ	100	100	100	ws*

Table 2: Test set performance averaged over all runs for each task and for the best combination of feature set and number of features, precision and recall having equal weight.
Key: all: ws/wspos/wspp/cw/cwpos/cwpp, cw*: cw/cwpos/cwpp, ws*: ws/wspos/wspp

tors are assigned proportionally to each class and deteriorates with the number of codebook vectors, a clear sign of overfitting.

LVQ achieves a performance ceiling of 100% precision and recall on nearly all tasks except for genre task A. The low average performance of IBL is due to bad results for $k = 1$; for higher k, IBL performs as well as LVQ. Overall, performance decreases with increasing number of features. IBL is rather robust regarding the choice of feature set. LVQ tends to perform better on data sets derived from both content and function words, with the exception of task A. Because of the ceiling effect, it almost never matters if the additional linguistic features are included or not. Recall is significantly better than precision for most tasks.

RIBL shows the greatest variation in performance. Although it performs fairly well, Tab. 2 shows differences of up to -5% on precision and -23% on recall. Overall, ws-based feature sets outperform cw-based ones. Performance declines sharply with the number of features. POS features almost always have a clear positive effect on recall (on average +28%, cw* and +16%, ws*), but an even larger negative effect on precision (-38%, cw* and -39%,ws*), which only shows for 500 and 1000 lemma features. Lemma and POS frequency information apparently conflict, with POS frequency leading to overgeneralization. Maybe semantic features describe the class boundaries more adequately. They may be covered implicitly in large vectors containing lemmata from that class. For 100 lemma features, where the representation is extremely sparse, we find that including POS information does indeed boost performance, especially for the two genre tasks, as we would have predicted.

5.4 CL Experiments

5.4.1 Procedure

In this set of experiments, RIBL and IBL were both evaluated using leave-one-out cross validation. The performance of LVQ is reported on the basis of ten-fold cross validation for reasons of computing time. Training and test sets were also constructed somewhat differently. The test set contained the same proportion of positive examples as the training set. If we had balanced the test set as above, this would have resulted in 4 pairs of sets instead of 10, and much smaller test sets, because some classes, such as L, are very small. This problem was not so grave for the LPE experiments because of the ceiling effect and the small size of the complete data set, therefore, we did not rerun the corresponding experiments. Furthermore, the number of codebook vectors for LVQ was now varied between 10, 50, 100, and 200 in order to take into account the increased training set sizes.

5.4.2 Results

The results on the larger corpus differ substantially from that on the smaller corpus. It is far easier to determine if a text belongs to one of the three major domains covered in a corpus than to assign a text to a minor domain which covers only 4% of the complete corpus. If the class itself is not considerably more homogeneous (with respect to the classifier used) than the rest of the corpus, this will be a difficult task indeed. Our results suggest that the classes were indeed not homogeneous enough to ensure reliable classification. The reason for this is that LIMAS was designed to be as representative as possible, and consequently to be as heterogeneous as possible. This explains why we never achieved 100% precision and recall on any data set again. In fact, results became much worse, and varied a lot depending mainly on the type of classifier and the task. Again, if classes are very inhomogeneous, any change in the way similarity between data items is computed can have strong effects on the composition of the neighbourhood, and the erratic behaviour observed here is a vivid testimony of this. We therefore chose not to present general summaries, but to document some typical patterns of variation.

Parameter settings: LVQ gives best results in terms of both precision and recall for even initialisation of codebook vectors, which makes sense because the number of positive examples has now become rather small in comparison to the rest of the corpus. A good codebook size appears to be 50 vectors.

	H		S	
	50	200	50	200
CW	65.2	33.6	42.24	47.15
CWPOS	65.2	29.5	42.24	47.15
FW	19.6	54	59.79	17.3
FWPOS	19.6	54	74.4	17.3
WSPOS	88.3	100	62.45	45.9
WS	56.6	68	62.45	45.9

Table 3: Average LVQ results (precision) for categories H and S, 50 codebook vectors, even initialization.

For RIBL, restricting the size of the relevant neighbourhood to 1 or 2 gives by far the best results in terms of both precision and recall, but not in terms of accuracy - the negative effect of false positives is too strong.

IBL is also sensitive to the size of the neighbourhood; again, precision and recall are highest for k=1. For this size, incorporating information gain into the distance measure leads to a clear decrease in performance.

Overall performance: Unsurprisingly, performance in terms of precision and recall is rather poor. Average LVQ performance under the best parameter settings in terms of precision and recall only improves on the baseline for two genres: H (baseline 78%, accuracy for feature set WSPOS 88%) and FL (feature sets CONT and CONTPOS, baseline 94%, accuracy 95%). Under matched conditions (same genre, same feature set, same number of features, optimal settings), IBL and RIBL both perform significantly worse than LVQ, which can interpolate between data points and so smooth out at least some of the noise. For example, IBL accuracy on task H is 69,1% for both WS and WSPOS, while accuracy on FL never much exceeds 92% and thus remains just below baseline. RIBL performs best on FL for condition CWPOS, but even then accuracy is only 90%.

Size of Feature Vector: The number of features used did not significantly affect the performance of IBL. For LVQ, both precision and recall decrease sharply as the number of features increases (average precision for 50 lemma features 29.5%, for 200 24.8%; average recall for 50 9.1%, for 200 7.1%). But this was not the case for all genres, as Tab. 3 shows. The categories H and S are chosen for comparison because they are the largest. For H, the precision under conditions CW and CWPOS decreases, all others increase; for S, it is exactly the other way around.

Composition of feature vectors: Another lesson of Tab. 3 is that the effect of the composition of the feature vectors can vary depending both on the task and on the size of the feature vector. The dramatic fall in precision for condition FWPOS, category S, shows that very clearly. Here, additional function word information has blurred the class boundaries, whereas for H, it has sharpened them considerably. Because of the large amount of noise in the results, we would be very hesitant to identify any condition as optimal or indeed claim that our hypotheses about the role of POS information or content vs. function words could be verified. However, what these results do confirm is that sometimes, comparing different representations might well pay off, as we have seen in the case of task H, where WSPOS indeed emerges as optimal feature set choice.

6 Conclusion

In this paper, we examined different linguistically motivated inputs for training text classification algorithms, focussing on domain- and genre-based tasks.

The most clear-cut result is the influence of the training corpus on classifier performance. If we want general-purpose classifiers for large genres or collections of genres, "small" representative corpora such as LIMAS will in the end provide too little training material, because the emphasis is on capturing the extent of potential variation in a language, and less on providing sufficient numbers of prototypical instances for text categorisation algorithms. In addition, genre boundaries are notoriously fuzzy, and if this inherent variability is compounded by sparse data, we indeed have a problem, as Sec. 5.4 showed. Therefore, further work into genre classification should focus on well-defined genres and corpora large enough to contain a sufficient number of prototypical documents. In our opinion, further investigations into the utility of linguistic features for textcategorization tasks should best be conducted on such corpora.

Our results neither support nor refute the hypotheses advanced in Sec. 2. However, note that in some cases, the additional non-content word information did indeed improve performance (cf. Tab. 3), so that such representations should at least be experimented with before settling on content words.

Acknowledgements

We would like to thank Stefan Wrobel, Thomas Portele, and two anonymous reviewers for their comments. All statistical analyses were conducted with R (http://www.ci.tuwien.ac.at/R). Oliver Lorenz added the POS tags to LIMAS.

References

D. Aha, D. Kibler, and M. Albert. 1991. Instance-based learning algorithms. *Machine Learning*, 6:37-66.

H. Bergenholtz and J. Mugdan. 1989. Zur Korpusproblematik in der Computerlinguistik. In I. Bátori, W. Lenders, and W. Putschke, editors, *Handbuch Computerlinguistik*. deGruyter, Berlin/New York.

B. Beutel. 1998. Malaga User Manual. http://www.linguistik.uni-erlangen.de/Malaga.de.html.

D. Biber. 1988. *Variation across Speech and Writing*. Cambridge University Press, Cambridge.

U. Bohnebeck, T. Horvath, and S. Wrobel. 1998. Term comparisons in first-order similarity measures. In *Proc. 8th Intl. Conf. Ind. Logic Progr.*, pages 65–79.

W. Daelemans, A. van den Bosch, and T. Weijters. 1997. IGTtree: Using trees for compression and classification in lazy learning algorithms. *AI Review*, 11:407-423.

W. Emde and D. Wettschereck. 1996. Relational instance based learning. In *Proc. 13th Intl. Conf. Machine Learning*, pages 122–130.

R.I. Forsyth and D. Holmes. 1996. Feature-finding for text classification. *Literary and Linguistic Computing*, 11:163–174.

R. Glas. 1975. Das LIMAS-Korpus, ein Textkorpus für die deutsche Gegenwartssprache. *Lingustische Berichte*, 40:63-66.

G. Herdan. 1960. *Type-token mathematics: a textbook of mathematical linguistics*. Mouton, The Hague.

D. Holmes. 1998. The evolution of stylometry in humanities scholarschip. *Literary and Linguistic Computing*, 13:111-117.

T. Joachims. 1998. Text categorization with Support Vector Machines: Learning with many relevant features. Technical Report LS-8 23, Dept. of Computer Science, Dortmund University.

J. Karlgren and D. Cutting. 1994. Recognizing text genres with simple metrics using discriminant analysis. In *Proc. COLING Kyoto*.

B. Kessler, G. Nunberg, and H. Schütze. 1997. Automatic classification of text genre. In *Proc. 35th ACL/8th EACL Madrid*, pages 32–38.

J. Klavans and Min-Yen Kan. 1998. Role of verbs in document analysis. In *Proc. COLING/ACL Montréal*.

T. Kohonen, J. Kangas, J. Laaksonen, and K. Torkkola. 1996. LVQ-PAK - the learning vector quantization package v. 3.0. Technical Report A30, Helsinki University of Technology.

I. Kononenko. 1994. Estimating attributes: Analysis and extensions of relief. In *Proc. 7th Europ. Conf. Machine Learning*, pages 171 - 182.

H. Kučera and W Francis. 1967. *Frequency analysis of English usage: lexicon and grammar*. Houghton Mifflin, Boston.

D. Lewis. 1992. Feature selection and feature extraction for text categorization. In *Proc. Speech and Natural Language Workshop*, pages 212–217. Morgan Kaufman.

C. Martindale and D. MacKenzie. 1995. On the utility of content analysis in author attribution: The Federalist. *Computers and the Humanities*, 29:259–270.

U. Pieper. 1979. *Über die Aussagekraft statistischer Methoden für die linguistische Stilanalyse*. Narr, Tübingen.

D. Ross and D. Hunter. 1994. μ-EYEBALL: An interactive system for producing stylistic descriptions and comparisons. *Computers and the Humanities*, 28:1–11.

G. Salton and M.J. McGill. 1983. *Introduction to Modern Information Retrieval*. McGrawHill, New York.

A. Schiller, S. Teufel, and C. Thielen. 1995. Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, IMS Stuttgart/Seminar f. Sprachwiss. Tübingen.

J. Swales. 1990. *Genre Analysis*. Cambridge University Press, Cambridge.

A. von der Grün. 1999. Wort-, Morphem- und Allomorphhäufigkeit in domänenspezifischen Korpora des Deutschen. Master's thesis, Institute of Computational Linguistics, University of Erlangen-Nürnberg.

Y. Yang and J. Pedersen. 1997. A comparative study on feature selection in text categorization. In *Proc. 14th ICML*.

Y. Yang. 1997. An evaluation of statistical approaches to text categorization. Technical Report CMU-CS-97-127, Dept. of Computer Science, Carnegie Mellon University.