The Feature Difference Coefficient: Classification Using Feature Distribution

Ulli Waltinger and Alexander Mehler

University of Bielefeld
Computer Science Department
Universitätsstrasse 25, 33615 Bielefeld, Germany
{ulli_marc.waltinger,alexander.mehler}@uni-bielefeld.de

Abstract. This paper presents a model of text classification using feature frequency distribution. The proposed algorithm offers not only sensitivity to linguistic but also to structure features and calculates a unified fingerprint for each category. Classification is done by finding the closest match to pre-learned models using a simple distance metric. The approach will be evaluated against three different classification scenarios. Language identification, text classification based on the Reuters corpus and web genre classification.

Key words: Text classification, N-Gram, SVM, Web Genre, Zipf's Law

1 Introduction

The process of text categorization has been intensively studied in the past. Besides others, approaches of machine learning techniques [1] have been proven to be adequate for an automatic classification scenario. In special, particularly implementations of support vector machines (SVM) [2] are consequently a common technique in approaching this issue. More precisely, feature selection [3, 4] and conversion to a binary input format is the most provoking and important point in using SVM. That is, the task of feature selection varies from the classification intention point of view. N-Gram based classification techniques have been also introduced for language identification [5], text categorisation [6, 7] and web genre identification [8] successfully. Thus, those approaches mainly focus on the task of feature selection of n-grams as a subsequence of textual constraints. [8] proposed a combination of structural and textual information in the process of feature selection and training afterwards a SVM for web genre classification. [6] evaluated the minimum and maximum threshold of features for text categorisation using only character n-grams of text. Our approach follows in principle the idea introduced by [5] using a distance metric of a descending ordered list of n-grams in order to judge the amount of overlap between two different n-gram frequencies. We continue the main idea by introducing different linguistic, textual and structural feature models as a fingerprint for different categories to be assigned. Doing this, we combine not only word form and string character information, but also part of speech and named entity specification. Regarding

structural constraints we comprise also the logical document structure as well as html attributes as prominent features.

The paper is structured as follows. A general overview of our algorithm is outlined in Section 2. We describe the task of automatic feature selection and building the category fingerprints. Section 2.2 will describe the classification scenario for classifying unknown text using our Feature Difference Quantity Classification approach. Section 4 will present the results of the experiments (Section 3) conducted using the Wikipedia, Reuters and 7-Web Genre corpus.

2 Methodology

Parsing text corpora by token frequency information, detailed information about the distribution of words can be gained. This follows Zipf's law [9], which states that only a few words are used frequently and the most are used infrequently. On the one side, frequent tokens will reflect the most common words of a language (function words as for example: *he, she, it*) on the other side these will be also domain specific words expressing the relevance of a text. Following this idea, we can argue that if two documents have an similar token frequency, both are related in some way. Adapting this approach to structural information, we assume that there might be also a similarity in the usage of structural relevant information as for example average sentence length, occurrences of grammatical features (e.g. frequency of nouns in a text) or structural features (e.g Html attributes or tags). In general, we argue that there are text types dominated by their structure as for example web documents, which contain additional information such as Html tags, attributes and hyperlinks. On the other side, traditional text documents might be vary in the usage of word class information and the usage of certain named entity information such as proper names of people or countries. Therefore, we are introducing different models which tie these information to a much more general fingerprint of one category to be classified.

2.1 Category Profile

In general, classification is done by using different corpus features. In this case, we differentiate between textual, linguistic and structural information for the process of automatic feature construction. We extract all predefined information out of a text corpus and build nine sub models associated to one category:

1. Token N-Grams: Occurrences of n-gram using a sequence of n word forms.
2. Letter N-Grams: Occurrences of n-gram using a sequence of n characters.
3. Token Frequency: Occurrence of a single word form.
4. Structure Length: Standard deviation of sentences, divisions, paragraph length.
5. HTML-Tag Frequency: Occurrences of n-gram using a sequence of n html tags based on the logical document structure only.
6. HTML-Attribute Frequency: Occurrences of n-gram using a sequence of

Feature	Frequency	Rank
the	634	1
to	477	2
f	398	3
m	378	4
th	245	5
...	...	..

Table 1. Feature Ranking of Letter N-Grams

n html attributes.
7. First-Last Tag Frequency: Occurrences of n first and last occurring html tags.
8. Part of Speech Tag Frequency: Occurrences of n-gram using a sequence of n Part of Speech Tags.
9. Named Entity Frequency: Occurrences of word forms classified as a Named Entity Category (e.g. Person, Country, Location, Time ...)

Each sub model consists of a ranked frequency distribution of certain assigned corpus features. That is, we merely parse each text assigned to one category and count the occurrences of the individual accounted features. In special, we read the incoming text split the raw data in html and text only representation and perform a tokenization, sentence detection, a part of speech tagging and named entity recognition. We used the system of [10] for the task of preprocessing. In a next step, corresponding n-gram information with parameter $n = 1$ to $n = 5$ are extracted. Each n-gram gets its own frequency counter.

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

We compute the frequency distribution tf of relevant features and order all in descending order for each sub model. Each feature gets a specific rank number (see Table 1 for a ranking of letter n-grams) according to its index position in the frequency list f_l. For example the rank number 1 is the feature with highest frequency, rank number $length(f_l)$ the feature with the lowest frequency information based on the training corpus. Note, features with the same frequency information have the same rank position in common. We did not perform a cutoff rank for the experiments, that is we used all extracted features for the task of classification. See Figure 1 for frequency plot of the html feature distribution of a weblogs and search pages.

2.2 Category Profile Ranking

After training each model for each target category we are in a position to process input documents in order to predict their unknown category. This is done by using the overlap index proposed by [5] which measures distances between

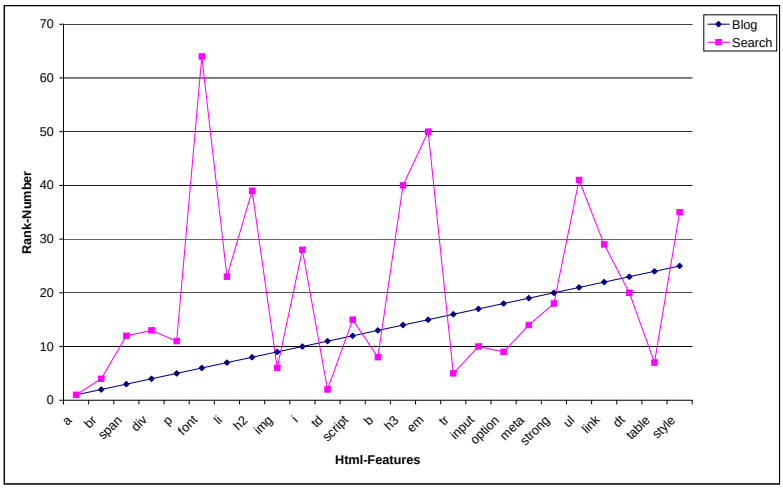

Fig. 1. Html-Features By Rank (Blog vs. Search Pages)

input documents based on the ranked profiles of their n-gram frequencies. In our case document models are separated into submodels which code separate feature rankings as input to categorization. Thus, our task is to compute the overlap of category- and document-related submodels per type of model enumerated in Section 2.1. In order to do that each input document x_m is processed by analogy to category profiles (see above). That way, document x_m becomes comparable to each category C_n by considering submodels. This comparison is done by calculating the distance d_{mnt} between the rank r_{mt} of feature t in the corresponding submodel of document x_m on the one hand and the rank r_{nt} of the same feature in the corresponding submodel of category C_n on the other:

$$d_{mnt} = \begin{cases} |r_{mt} - r_{nt}| & t \in m \wedge t \in n \\ Max & t \notin m \vee t \notin n \end{cases} \tag{2}$$

If feature t is not ranked in the model of category C_n the maximum distance Max is assumed. Max is set to the highest rank of the operative submodel augmented by 1. Examples of two submodels are given in Table 2 – which shows the ranking of letter-based n-grams – and in Table 3 which demonstrates the ranking of PoS features. This process is repeated for all features occurring in the submodels of input document x_m. By summing up all distances d_{mni} for a given submodel $\mathbb{M}^k$ for each of the $1 \leq i \leq t$ features of x_m we get an overall index $\mathbb{M}^k_{mn}$ of the

Model	Frequency	Rank (r_m)	Document	Frequency	Rank (r_d)	Distance
the	634	1	tn	223	1	4
to	477	2	to	212	2	0
f	398	3	m	134	3	1
m	378	4	as	98	4	max
th	245	5	f	87	5	2
...	...	..	...	...	...	...
Distance (D_{md})						7+max

Table 2. Feature Distance Ranking of Letter N-Grams

"compatibility" of category C_n and document x_m *in terms of submodel* $\mathbb{M}^k$:

$$\mathbb{M}^k_{mn} = \sum_{i=1}^{t} d_{mni} \tag{3}$$

The larger $\mathbb{M}^k_{mn}$, the less compatible C_n and x_n in terms of the submodel $\mathbb{M}^k$, the smaller the probability that x_n is correctly categorised by C_n when exploring solely $\mathbb{M}^k$ as a source of information. As we deal with nine different such information sources, that is, with nine different types of submodels we introduce a parameter δ_k for biasing the impact of $\mathbb{M}^k_{mn}$ on the overall categorisation. That is, we compute:

$$C_{mn} = \sum_{k=1}^{9} \delta_k \cdot \mathbb{M}^k_{mn} \tag{4}$$

where

$$\sum_{k=1}^{9} \delta_k = 1 \tag{5}$$

This gives us an index of the compatibility of document x_m and category C_n: the smaller C_{mn} the more features are similarly ranked for this document and category, the higher the probability that x_m is correctly categorised by C_n. The last step is to select those category C_{fin} whose feature profile is most compatible with that of document x_m, that is

$$C_{\text{fin}} = \arg\min_{C_m \in \mathbb{C}} \{C_{mn}\} \tag{6}$$

where $\mathbb{C}$ is the set of target categories.

A note on the parameter δ: δ is used to vary the impact of textual, structural and lexical sources of information as represented by corresponding submodels. That is, if we want to focus, e.g., more on structural information, δ is lowered in the case of the submodels `Structure Length`, `HTML-Tag Frequency` and `HTML-Attribute Frequency`. This is done in order to give the corresponding distance sums a smaller impact on the final categorisation. Building the sum of all weighted sub model distances of one category, we gain the membership value C_m of an unknown document to one category. This calculation is repeated for

Model	Frequency	Rank (r_m)	Document	Frequency	Rank (r_d)	Distance
NN	3216	1	NN	86	1	0
NNP	2510	2	IN	24	2	1
IN	2397	3	NNP	13	3	1
CARD	2026	4	CARD	9	4	0
NNS	1568	5	NNP_NNP	7	5	max
...	...	..	...	...	...	...
Distance (D_{md})						2+max

Table 3. Feature Distance Ranking of POS-Tags

Model	Frequency	Rank (r_m)	Document	Frequency	Rank (r_d)	Distance
a	14218	1	table	45	1	max
br	10296	2	h6	5	2	max
span	5065	3	abbr	3	3	max
div	3965	4	a	3	4	3
p	2587	5	dt	1	4	max
...	...	..	...	...	...	...
Distance (D_{md})						3+(4*max)

Table 4. Feature Distance Ranking of HTML-Tags

all trained models and retrieving therefore all values for each trained category. In order to assign a category to the document we are building the minimum of all C_m. That is, the category with the lowest distance value is afterwards picked as the model C_s that corresponds best to the to classified document.

3 Experiment

In order to evaluate our proposed approach, we split our experiments into three different sections. In first place, using the Feature Difference Coefficient (FDC) for language identification. Second, doing experiments with the Reuters Corpus as an example for the standard classification evaluation. An third, we used our approach for web genre classification. Each outcome is evaluated against the representative baseline results of *SVM*, *Bayes*, *Rochio* or *rulebased* approaches. We used a quite divers experiment setting in order to show the broad range of application of our approach.

3.1 Language Classification

The task of language identification has already proven to be solved on a satisfied basis using only token n-grams. Nevertheless, we conducted a classification experiment based upon 22 different languages and tried to predict the minimum amount of characters (less then 300 characters). As a training corpus we used 1000 articles and as the evaluation corpus another 1000 articles for each language randomly selected from the *Wikipedia* project. Further, we used a sliding

window from 10 up to 300 chars for a random extractor from the evaluation corpus. For each language and each maximum allowed char length we classified 100 different strings and calculated the F1-Measure. The F1-Measure is the weighted harmonic mean of precision and recall defined as:

$$F_{measure} = 2 \cdot (\text{precision} \cdot \text{recall})/(\text{precision} + \text{recall}) \tag{7}$$

We used no structure information of the *Wikpedia* article and set no feature limit during training or classification.

3.2 Text Classification

The empirical evaluation for the standard text classification scenario is done using the Reuters-Collection, which is a famous collection for text classification. We used the *ModApte split* of the Reuters-21578 dataset compiled by David Lewis[1], which includes only human assessed documents and comprises the collection into training and testing documents. Starting from this, we extracted all documents assigned uniquely to one of the test largest topic categories. This lead to an overall corpus of 9133 documents. 6552 documents were used to train the categories and 2581 documents were used to test the classification. As the competitive approaches in text classification we used the results reported from [2], [11] and [12]. Different to the language identification we added linguistic features to our classification models(PoS-tag and named entity information).

3.3 Hypertext Classification

As an third collection, we used the seven web genre corpus proposed from [13]. This collection covers hypertext types as *web logs, e-shops, FAQs, front pages, listings, personal home pages* and *search pages*. The corpus consists of downloaded web documents, therefore structure, linguistic *and* text information is present. The overall corpus comprises 1400 web documents of which 700 were used to build the web genre profiles and 700 werde used to to test the classification. As the golden standard we compare our approach with the results reported from [13] using SVM and an inferential model. Accuracy is used as the statistical measure:

$$\text{Accuracy} = \frac{\text{true positives} + \text{true negatives}}{\text{true positives} + \text{false positives} + \text{false negatives} + \text{true negatives}} \tag{8}$$

4 Results

Overall, the experiments show quite promising results. Regarding language identification (see Table 5) we gain an overall F1-Measure of 0.973 with a minimum

[1] http://www.daviddlewis.com/resources/testcollections/reuters21578/

Language (21)	10 chars	20 chars	50 chars	80 chars	100 chars	300 chars
Chinese	0.924	0.969	0.979	0.989	0.979	1.000
German	0.830	0.974	1.000	1.000	1.000	1.000
English	0.802	0.924	0.974	1.000	1.000	1.000
Finnish	0.876	0.895	0.969	0.989	0.979	1.000
French	0.843	0.958	1.000	0.974	0.984	1.000
Greek	0.989	0.994	1.000	1.000	1.000	1.000
Italian	0.561	0.692	0.692	0.742	0.802	1.000
Japanese	0.787	0.75	0.843	0.830	0.850	1.000
Korean	0.901	0.994	1.000	1.000	1.000	1.000
Croatian	0.802	0.901	0.984	0.994	1.000	1.000
Latin	0.581	0.742	0.952	0.974	1.000	1.000
Dutch	0.630	0.802	0.963	0.989	0.979	1.000
Norwegian	0.717	0.882	0.969	0.989	0.994	1.000
Polish	0.734	0.823	0.979	0.989	0.952	1.000
Portuguese	0.882	0.918	0.984	0.994	0.989	1.000
Rumanian	0.850	0.947	1.000	0.994	1.000	1.000
Russian	0.882	0.930	0.989	0.994	1.000	1.000
Swedish	0.816	0.924	1.000	0.994	0.994	1.000
Serbian	0.924	0.994	1.000	1.000	1.000	1.000
Spanish	0.684	0.809	0.924	0.974	0.989	1.000
Turkish	0.895	0.994	0.994	1.000	1.000	1.000
Czech	0.850	0.936	1.000	1.000	1.000	1.000
Average	0.807	0.897	0.963	0.973	0.976	1.000

Table 5. Language Identification with FDC (F-Measure)

amount of 50 chars as an unknown input string. Expanding this input up to 300 chars we constantly compute an F-Measure of 1.00. This results confirm the results of previous studies [5]. Using the same classification application for the experiment of the Reuters Corpus (see Table 6) we clearly outperform existing approaches as *Bayes Rocchio*, *kNN* and the *Trees*-Classification. With the current implementation we are almost as good as the SVM implementation gaining an F-Measure of 0.84 compared to 0.87. Using a web genre corpus for training our model better results can be achieved. We clearly outperform the Baseline (Bayes). In addition, with an average accuracy of 0.93 we also outperform all other state-of-the-art methods including the inferential model approach of [14] and the SVM implementation. The performance of the proposed method clearly indicates the usefulness for the application of web genre classification. Especially the possibility in measuring the similarity the usage of structural information of text types contributes to this. In addition, the advantage of our approach is that no binary feature conversion is needed for the task of classification. Building the models is easy and simple since only raw input text is needed for training. All features are automatically extracted and rated in a ranking manner.

Classes (9)	Bayes	Rocchio	kNN	Trees	SVM	FDC
earn	96	93	97	98	98	98
acy	88	65	92	90	94	97
money-fx	57	47	78	66	75	84
grain	79	68	82	85	95	70
crude	80	70	86	85	89	90
trade	64	65	77	73	76	77
interest	65	63	74	67	78	75
ship	85	49	79	74	86	64
wheat	70	69	77	93	92	-
corn	65	48	78	92	90	-
coffee	-	-	-	-	-	97
money-supply	-	-	-	-	-	85
Average	75(82)	64(65)	82(82)	82(88)	87(92)	84(91)

Table 6. Result Reuters Classification [F-Measure (Accuracy)]

Classes (9)	Bayes	Inferential Model	SVM	FDC
blogs	92	91	96	98
eshops	76	83	88	91
faqs	67	88.5	94.5	98
front pages	98	97	100	100
listings	29	77.5	80	87
personal home	27	77	79	92
search pages	82	88	85	91
Average	67	86	89	93

Table 7. Result Hypertext Type Classification (Accuracy)

5 Conclusion

This paper presented the feature difference coefficient, a model of text classification using feature frequency distribution. The proposed algorithm offers not only sensitivity to linguistic but also to structure features and calculates a unified fingerprint for each category. Classification is done by finding the closest match to pre-learned models using a simple distance metric. The approach was evaluated against three different classification scenarios. Language identification, text classification based on the Reuters corpus and web genre classification.

Acknowledgement

Financial support of the German Research Foundation (DFG) through the Excellence Cluster 277 *Cognitive Interaction Technology*, the Research Group 437 *Text Technological Information Modeling* and the LIS-Project *P2P-Agents for Thematic Structuring and Search Optimisation in Digital Libraries* at Bielefeld University is gratefully acknowledged.

References

1. Sebastiani, F., Ricerche, C.N.D.: Machine learning in automated text categorization. ACM Computing Surveys **34** (2002) 1–47
2. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: European Conference on Machine Learning (ECML), Berlin, Springer (1998) 137–142
3. Taira, H., Haruno, M.: Feature selection in svm text categorization. In: AAAI '99/IAAI '99: Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence, Menlo Park, CA, USA, American Association for Artificial Intelligence (1999) 480–486
4. Gabrilovich, E., Markovitch, S.: Text categorization with many redundant features: Using aggressive feature selection to make svms competitive with c4.5. In: Proceedings of The Twenty-First International Conference on Machine Learning, Banff, Alberta, Canada, Morgan Kaufmann (2004) 321–328
5. Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. (1994) 161–175
6. Mansur, M., UzZaman, N., Khan, M.: Analysis of n-gram based text categorization for bangla in a newspaper corpus. In: 9th International Conference on Computer and Information Technology (ICCIT 2006), Dhaka, Bangladesh (2006)
7. Urnkranz, J.F.: A study using n-gram features for text categorization
8. Kanaris, I., Stamatatos, E.: Webpage genre identification using variable-length character n-grams. In: ICTAI '07: Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence - Vol.2 (ICTAI 2007), Washington, DC, USA, IEEE Computer Society (2007) 3–10
9. Zipf, G.K.: Human Behavior and the Principle of Least Effort. Addison-Wesley (Reading MA) (1949)
10. Waltinger, U., Mehler, A.: Web as preprocessed corpus: Building large annotated corpora from heterogeneous web document data. In preparation (2009)
11. Yang, F.L.Y.: A loss function analysis for classication methods in text categorization (2003)
12. Dumais, S., Platt, J., Heckerman, D., Sahami, M.: Inductive learning algorithms and representations for text categorization. In: CIKM '98: Proceedings of the seventh international conference on Information and knowledge management, New York, NY, USA, ACM Press (1998) 148–155
13. Santini, M., Power, R., Evans, R.: Implementing a characterization of genre for automatic genre identification of web pages. In: Proceedings of the COLING/ACL on Main conference poster sessions, Morristown, NJ, USA, Association for Computational Linguistics (2006) 699–706
14. Santini, M.: Identifying genres of web pages. In: Proceeding of TALN 2006 (Traitement Automatique des Langues Naturelles. (2006)