Detecting Documents with Complaint Character

Sebastian Ebert
University of Munich (LMU),
Munich, Germany
ebert@cis.uni-muenchen.de

Benjamin Adrian
Insiders Technologies GmbH
Kaiserslautern, Germany
B.Adrian@insiders-technologies.de

Abstract

Recognizing complaint documents as early and as fast as possible is a worthwhile goal for companies. In this paper we present an analysis showing the complexity of this practically relevant problem. Therefore, we define the task and its challenges and investigate statistical methods for automated Complaint Detection in incoming text documents. Two different approaches for handling complaint documents are presented. First, we analyze various term weightings in a standard bag-of-words approach. Second, we show the effect of feature engineering techniques known from Natural Language Processing. The results on four German and one English corpora show that already a linear classifier achieves valuable results and is competitive to more sophisticated methods in most cases.

1 Introduction

Complaints express a person's dissatisfaction and usually contain displeasure, anger, or other negative mood, since the sender is unhappy with some circumstance. Triggering events may be a company's products or services. Complaints are valuable for companies. If handled appropriately, i.e., if there is a good working management of complaints, both customer as well as the company will win satisfaction. The customer receives help and the company has a more satisfied customer. Additionally, complaints are opportunities, which can point out general problems. Fixing such issues improves the quality of products and services and reaches many customers at once.

Many of today's companies detect and handle complaints in the following way. A writing, e.g., a letter or an email, is received, scanned and forwarded to a document analysis system. Such a system extracts the text from the scanned writing and converts it into digital text. Then, information is extracted from the writing, which helps classifying it into company specific document categories, like car insurance or health insurance. Often the document is forwarded to a specific employee group based on this category. Such a group reads the text and if it is a complaint, either handles it herself/himself or forwards it to specialized complaint team. As a consequence, a complaint is handled only when an employee has recognized it. Since complaining customers are likely to cancel a company's services there is a need for prioritized handling of complaint documents. An automated Complaint Detection (CD) system is able to detect complaints even before an employee had to read a single document. This will dramatically reduce a company's reaction time.

In this paper we deal with the automatic detection of complaint documents in incoming mail. We investigate several Machine Learning (ML) methods on their suitability for this task. Using automated CD combines the benefits of complaint management, e.g., prioritized handling of complaints, with the faster approach of computer-supported detection of complaint documents.

The major challenges arising from complaints are the following:

Domain dependency Every company or even every department in a company needs to define what a complaint is. Thus, the definition can be totally different from department to department. Such differences lead to a tight domain dependency. We present a trainable method that can be adapted to different domains.

Consistent guidelines Instructing employees to recognize complaints is a difficult task, because there must be consistently and precisely formulated decision guidelines. Otherwise, one employee might say it is a complaint, another one may say it is not. Our statistical method ensures that a consistent definition of complaints is enforced and human error is eliminated as a source of inconsistencies.

Amount of documents The amount of incoming documents in a company can be higher than 1 million a day. Here, even a very low relative rate of misclassified documents leads to a high absolute number of not found complaints or writings wrongly declared as complaints. The former case vanishes the advantage of prioritized complaint handling. Furthermore, a low false negative rate is of particular importance in CD because reliable detection of the first complaint about a new problem and a quick elimination of the root cause can prevent a large number of subsequent complaints about the same problem as well as the high cost of losing dissatisfied customers and undoing damage that has already been done. A high false positive rate is undesired, because companies fear too much additional manual reclassification effort for their complaint team.

2 Related Work

Generally, detecting complaint documents is a classification task. In opposite to other classification tasks, e.g., topic classification, we have only two classes, namely complaints and non-complaints. The documents in either class do not share a certain topic. Instead, the similarity of all complaint documents is that the sender is unsatisfied with some circumstance; reasons are quite diverse. The diversity within the non-complaint documents is even larger. They can deal

with any topic, product, or service. Documents can be for example invoices, offers, or notification letters. The only thing these documents have in common is that the sender does not complain.

We belief that CD is similar to the task of Sentiment Analysis (SA). It is likely that complaint documents are written in a negative way. Much research in SA has been carried out in the movie domain. For example, Pang et al. [2002] classified the polarity (negative or positive) of movie reviews using ML algorithms, namely Naïve Bayes, Maximum Entropy, and Support Vector Machine (SVM). The authors studied the effect of term weighting schemes (binary, term frequency), bigrams, and the position of terms in a review on the polarity classification performance. In our work we carry out a more thorough research on term weighting schemes and also evaluate the use of trigrams, which allow to find longer structures. Furthermore, we look at several additional feature selection and feature extraction methods, not performed by Pang et al. [2002]. Lastly, our experiments are carried out on four German corpora and a larger movie review corpus.

The subfield of subjectivity classification deals with the distinction between subjective and objective texts [Wiebe, 2000]. Intuitively, non-complaints are always objective, like invoices, orders, etc.. However, subjective texts that are non-complaints are common, e.g., praises or accident reports in insurance companies. Such texts contain many polar words and often subjective language, but are no complaints. Moreover, somebody can complain without using subjective or polar speech. Consider the example sentence: "Why do you require 2 months for responding to my letter?" There is no explicit sentiment, i.e., a sentiment detector would probably classify it as a *neutral* sentence. Nevertheless, the sender is unhappy with the fact that nobody took care of her/his letter.

3 Term Weightings

The task of classifying a single document as being either a complaint or a non-complaint is a typical example of Text Classification (TC). In TC a given text document is assigned to one or more predefined classes [Sebastiani, 2002]. In this work, we formalize CD as a binary TC task, where the possible categories are *complaint* c_c and *non-complaint* c_n.

Documents are represented as bag-of-words: $d = \begin{bmatrix} w_1 & \dots & w_{|\mathcal{V}|} \end{bmatrix}^T$, where w_t is the weight of term t in this document and $\mathcal{V}$ is the vocabulary of all possible terms. A term weight is a numerical value that is assigned to a term. Salton and Buckley [1988] introduced a notation for term weights for their SMART retrieval system. This notation leads to a general definition for term weights: $w_{td} = f_t * f_c * f_n$, where the term weight for term t in document d consists of three factors: a *term frequency component* f_t, a *collection frequency component* f_c, and a *normalization component* f_n[1]. Table 1 lists the used components with their SMART notation and their computation.

For example, *txx* means that the number of occurrences of term t in document d, i.e., $\#(t, d)$ [Sebastiani, 2002] is taken solely as term weight. A very common term weighting in Information Retrieval (IR) is *tf-idf*, i.e., tfx, that accounts for the distribution of a given term t over the entire document corpus. df_t is the document frequency and counts in how many document the term occurs [Sebastiani, 2002]. In order to account for the document length, the term weight can be normalized by cosine normalization as presented in [Salton and Buckley, 1988]. A promising new term weighting for SA called *delta idf* (Δidf) was introduced by Martineau and Finin [2009], which instead of calculating the idf based on all documents, considers the idf values for the both classes separately and uses their difference: $\Delta idf_t = idf_{tc} - idf_{tn} = \log \frac{N_c * df_{tn}}{df_{tc} * N_n}$, where N_c and N_n represent the number of documents in the complaint and non-complaint class, respectively. df_{tc} and df_{tn} denote the document frequency of term t in the corresponding class. Paltoglou and Thelwall [2010] integrated Δidf into the SMART notation and created the so-called *smoothed* Δidf (Δf' in Table 1), which handles terms that occur in only one of the two classes.

	notation	computation
t_t	b (binary)	$b_{td} = \begin{cases} 1 & \text{if } t \text{ occurs in } d \\ 0 & \text{otherwise} \end{cases}$
	t (term freq.)	$tf_{td} = \#(t, d)$
	l (log)	$l_{td} = \log(1 + tf_{td})$
t_c	x (none)	1.0
	f (inv. doc. freq.)[2]	$idf_t = \log \frac{N}{1+df_t}$
	Δf' (smooth. Δidf)	$\Delta idf_t = \log \frac{N_c * df_{tn} + 0.5}{df_{tc} * N_n + 0.5}$
t_n	x (none)	1.0
	c (cosine)	$c_d = \frac{1}{\sqrt{\sum_{t=1}^{\lvert\mathcal{V}\rvert} w_{td}^2}}$

Table 1: SMART notation of weighting schemes

We evaluate all combinations of f_t, f_c, and f_n, which results in a total of 18 possible weighting schemes for a single corpus.

4 Feature Engineering

A document corpus represented as bag-of-words can contain millions of terms. Many classifiers cannot handle this amount of features, because they do not scale well [Sebastiani, 2002]. Furthermore, many algorithms are prone to overfitting if there are many features. Finally, the more features there are, the longer the training (and for some algorithms also the classification) takes. Therefore, a common approach in many Natural Language Processing (NLP) tasks is to reduce the number of features. We investigate the influence of three approaches, stemming, stop-word removal and Principal Component Analysis (PCA).

To perform stemming we use Snowball, a language created for writing stemming algorithms [Porter, 2001]. The English experiments are performed with the Snowball implementation of the Porter algorithm [Porter, 1980]. The German experiments are carried out with Snowball's German stemming algorithm.

In order to see the influence of stop-word removal, we use the German and English stop-word lists provided by

[1] The SMART notation actually consists of two triples: *ddd.qqq*, where *ddd* is the document representation and *qqq* is the query representation. We have no explicit queries and thus neglect the second triple.

[2] We want to consider new words from the test set and therefore use this version of idf instead of the common form $\log \frac{N}{df_t}$ in order to avoid division by zero.

corpus	compl.	non-compl.	no. of words
liability	55	170	6,039
car	1,088	2,610	66,961
damage 1	373	989	34,674
damage 2	372	865	31,461
IMDb	1,000	1,000	38,911

Table 2: Corpora statistics

the Snowball project. The English list contains 174 stop-words. The German list comprises 231 stop-words.

Another technique for reducing the number of features we investigate is PCA. It is an unsupervised technique that calculates a transformation T that transforms the high dimensional document term matrix M into a lower dimensional space M': $M' = TM$. Since the number of dimensions $m \ll |\mathcal{V}|$ the problem of high dimensionality is tackled.

Using only single words as features, as we have done so far, has a serious drawback. It neglects the position of terms and their context entirely. A common technique to incorporate the context of words are n-grams [Manning and Schütze, 2000]. In the experiments we use bigrams ($n = 2$) and trigrams ($n = 3$).

5 Experiments

We performed all presented techniques on four German corpora and one English corpus. The four German corpora are real data from real customers[3]. They were collected in four different German insurance company departments from daily incoming mail. The departments are liability insurance (*Liability*), car insurance (*Car*), and two different departments dealing with insurances against damage (*Damage 1* and *Damage 2*). The corpora consist of incoming paper letters or faxes. Each document ran through a typical image conversion pipeline with (i) digitizing the image, (ii) cleaning it in several preprocessing steps, and (iii) running an Optical Character Recognition (OCR) to retrieve machine readable text. The preprocessing of all digital text documents consists of lowercasing and tokenization. Every document was labeled as complaint or non-complaint by an employee of the respective department. Table 2 lists the number of complaints and non-complaints in the corpora after filtering out duplicates and documents that per se can never be a complaint, e.g., invoices. Additionally, the number of distinct words is shown. The distribution of text lengths is very similar for complaints and non-complaints.

In order to measure the difficulty of this the CD task we asked two outside parties to manually label 50 randomly chosen documents from the Car corpus (25 complaints, 25 non-complaints). Both persons were asked to label each document with either complaint or non-complaint according to their own understanding of a complaint. The two raters agreed in only $32/50$ documents ($\kappa = 0.28$), which shows the complexity of this problem and the need for *consistent guidelines*.

Since we assume that CD is similar to the field of SA, we use another corpus that is well-known in this domain. This corpus called *polarity dataset v2.0*, was introduced Pang and Lee [2004]. We refer to this document collection as *IMDb*, because it comprises 2000 movie reviews that were automatically extracted from the Internet Movie Database (IMDb) and labeled as being positive or negative. The corpus statistics are listed in Table 2. For the sake of simplicity, we treat the positive class as complaint and the negative class as non-complaint, in order to have a consistent class naming.

For classification we use the SVM implemented in libSVM from Chang and Lin [2011] with a linear kernel and default parameters. To obtain the SVM performance we perform 10-fold cross validation and average the final results to an overall performance. We measure precision, recall and F_1 for the complaint class, since we want to focus on complaints.

6 Results

There are three weighting schemes that produce the highest F_1 on at least one of the corpora. Due to the large number of combinations we only report resuts for these three weighting schemes. The configurations are: *bxx*, *tΔf'c*, and *bfc*. Table 3 lists the precision, recall, and F_1 results.

The term weighting bxx has achieved the best results on 3 out of 5 corpora with a difference of up to 10 % (Damage 1) to the second best weighting, although it is the most simple feature weighting. Bfc has a very positive effect on precision compared to bxx on all corpora. Thus, if the rate of False Positives (FPs) must be kept small, it is a better term weighting than a binary representation.

In our experiments, all combinations using the new Δidf weighting have often led to lower results than bxx. Even the best combination tΔf'c has shown inferior performance.

Although there are some differences in the performances depending on the corpus, the differences in F_1 performance have not been statistically significant for $p = 0.05$[4]. We conclude that there is no benefit computing complex weightings like Δidf, because binary weights already achieve good results. Therefore, we use bxx as the baseline for further investigations.

All dimensionality results were achieved using the bxx weighting scheme. They are listed in Table 4.

Stemming and stop-word removal have led to improved performance on only one corpus each (stemming: Damage 2, stop-word: IMDb). On all other corpora, the performance has been inferior. However, the differences have not been statistically significant. We do not recommend either of the two techniques.

In an optimal case, PCA strongly reduces the number of required features and still maintains the same performance. We have chosen the number of principal components in order to keep 95 % of the data's variance. For Liability this is 164 principal components (reduction of features by 97.3 %), for Car 2,372 (96.5 %), for Damage 1 957 (97.2 %), for Damage 2 874 (97.2 %), and for IMDb 1,439 (96.3 %). This a dramatic decrease in dimensionality. As Table 4 shows, performing PCA has not lowered the performance by much. The losses in F_1 have not been significant. Thus, PCA is very well suited to reduce the feature space and therefore reduce noise.

Using bigrams has resulted in a better F_1 performance on Car and IMDb. On the other corpora, the performance declined. Using trigrams could only improve the result on

[3] Due to privacy reasons this data may not be published.

[4] We performed a Friedman test with Holm's test as post-hoc test, following Demsar [2006].

	Liability			Car			Damage 1			Damage 2			IMDb		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
bxx	.83	.69	.75	.81	.75	.78	.75	.64	.69	.87	.84	.86	.85	.85	.85
bfc	.93	.25	.40	.89	.64	.75	.92	.36	.52	.93	.77	.84	.88	.88	.88
tΔf'c	.90	.67	.77	.76	.72	.74	.79	.47	.59	.86	.84	.85	.77	.81	.79

Table 3: Term weightings results

	Liability			Car			Damage 1			Damage 2			IMDb		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
bxx baseline	.83	.69	.75	.81	.75	.78	.75	.64	.69	.87	.84	.86	.85	.85	.85
stemming	.80	.67	.73	.80	.74	.77	.75	.64	.69	.88	.85	.86	.85	.86	.85
stop-word	.94	.56	.70	.79	.71	.75	.77	.60	.68	.87	.78	.82	.87	.85	.86
PCA	.84	.67	.75	.79	.73	.76	.74	.63	.68	.85	.83	.84	.85	.84	.84
2-grams	.84	.56	.67	.84	.75	.79	.84	.58	.69	.88	.80	.84	.88	.86	.87
3-grams	.83	.36	.51	.84	.72	.77	.84	.53	.65	.89	.77	.82	.89	.86	.88

Table 4: Feature engineering results

the IMDb corpus. This finding suggests, that n-grams cannot appropriately capture the context that is necessary to classify complaints.

7 Conclusion

In this paper we have introduced the topic of CD. We have argued that complaints are very important for companies as well as for customers.

As a first step in our research, we have shown that binary term representation has delivered as good results as more sophisticated methods or even better and their computation is both, easy and fast. But, if the system's FP rate is of importance and many documents are being misclassified as complaints, bfc should be preferred, because its precision results have generally been higher. Despite these results, the independence assumption that the unigram model makes is clearly wrong and in our case seems to be unable to capture complaints entirely. But also the use of n-grams, which consider more context, has not helped. Therefore, for the classification of complaint documents we need more linguistic knowledge, e.g., in terms of word polarities or discourse structures.

Using stemming or stop-word removal has not been beneficial, they have resulted in poorer results. Instead, PCA is well suited for drastically reducing the feature space (between 96.3 % and 97.3 %), while maintaining nearly equal results to those of the baseline. We conclude that other feature selection approaches may also be helpful in finding good complaint specific features.

This work is the basis for further analysis of complaint documents. In a next step we will investigate the usage of sentiment lexicons, which allow the incorporation of word polarities in the classification task.

Acknowledgement

We like to thank Insiders Technologies GmbH for providing support for this work and Hinrich Schütze for proof reading the paper.

References

[Chang and Lin, 2011] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27, 2011.

[Demsar, 2006] Janez Demsar. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[Manning and Schütze, 2000] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 2000.

[Martineau and Finin, 2009] Justin Martineau and Tim Finin. Delta TFIDF: An Improved Feature Space for Sentiment Analysis. In *Proceedings of the 3rd International Conference on Weblogs and Social Media*, 2009.

[Paltoglou and Thelwall, 2010] Georgios Paltoglou and Mike Thelwall. A Study of Information Retrieval Weighting Schemes for Sentiment Analysis. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1386–1395, 2010.

[Pang and Lee, 2004] Bo Pang and Lillian Lee. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 271–278, 2004.

[Pang *et al.*, 2002] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs Up?: Sentiment Classification Using Machine Learning Techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing - Volume 10*, pages 79–86, 2002.

[Porter, 1980] Martin F. Porter. An algorithm for suffix stripping. *Program: Electronic Library and Information Systems*, 14(3):130–137, 1980.

[Porter, 2001] Martin F. Porter. Snowball: A language for stemming algorithms, 2001. Access date: 07/16/2012.

[Salton and Buckley, 1988] Gerard Salton and Christopher Buckley. Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[Sebastiani, 2002] Fabrizio Sebastiani. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[Wiebe, 2000] Janyce M. Wiebe. Learning Subjective Adjectives from Corpora. In *Proceedings of the 17th National Conference on Artificial Intelligence*, pages 735–741. AAAI Press / The MIT Press, 2000.