Expert Systems with Applications 40 (2013) 4687–4696

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Using micro-documents for feature selection: The case of ordinal text classification ☆

Stefano Baccianella, Andrea Esuli, Fabrizio Sebastiani *

Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, 56124 Pisa, Italy

ARTICLE INFO

Keywords:
Text classification
Supervised learning
Ordinal regression
Feature selection

ABSTRACT

Most popular feature selection methods for text classification such as information gain (also known as "mutual information"), chi-square, and odds ratio, are based on binary information indicating the presence/absence of the feature (or "term") in each training document. As such, these methods do not exploit a rich source of information, namely, the information concerning how frequently the feature occurs in the training document (*term frequency*). In order to overcome this drawback, when doing feature selection we logically break down each training document of length k into k training "micro-documents", each consisting of a single word occurrence and endowed with the same class information of the original training document. This move has the double effect of (a) allowing all the original feature selection methods based on binary information to be still straightforwardly applicable, and (b) making them sensitive to term frequency information. We study the impact of this strategy in the case of ordinal text classification, a type of text classification dealing with classes lying on an ordinal scale, and recently made popular by applications in customer relationship management, market research, and Web 2.0 mining. We run experiments using four recently introduced feature selection functions, two learning methods of the support vector machines family, and two large datasets of product reviews. The experiments show that the use of this strategy substantially improves the accuracy of ordinal text classification.

1. Introduction

Text management tasks such as text search, text clustering, and text classification, are usually tackled by representing the textual documents in vectorial form. The resulting vector spaces are always characterized by a high dimensionality (often in the range of the tens, sometimes hundreds of thousands dimensions), since words (or word stems) are normally used as features, and since several tens of thousands of them occur in any reasonably-sized document space.[1] This very high dimensionality is not terribly problematic in text search, where the most fundamental operation (computing the distance between two vectors in the vector space) can be performed quickly, thanks to the sparse nature of the two vectors. It is instead problematic in other tasks involving supervised or unsupervised learning, such as text classification or clustering.

For instance, many supervised learning algorithms often used for text classification, such as neural networks, do not scale well to large numbers of features, and even the learning algorithms that do scale well have a computational cost at least linear in the dimensionality of the vector space. While this negatively impacts on efficiency, accuracy suffers too, since if the ratio $|Tr|/|W|$ of the number $|Tr|$ of training examples to the number $|W|$ of features is low, overfitting may occur, which typically leads to suboptimal classification accuracy.

For all these reasons, several techniques for reducing the dimensionality of a vector space in text learning tasks have been investigated, the main one being *feature selection* (see e.g. Forman, 2007; Huan & Hiroshi, 2007; Isabelle & Elisseeff, 2003; Maghsoodi & Homayounpour, 2011; Mengle & Goharian, 2009; Yang & Pedersen, 1997). This latter consists in identifying a subset $S \subset W$ of the original feature set W such that $|S| \ll |W|$ (with $\xi = |S|/|W|$ being called the *reduction level*) and such that S reaches the best compromise between (a) the efficiency of the learning process and of the classifiers (which is, of course, inversely proportional to $|S|$), and (b) the accuracy of the resulting classifiers.

The idea that underlies feature selection is, of course, that the most informative features must be retained in S while the least informative ones must be discarded. Here, "informative" actually means *discriminating*, in the sense that a feature t_k is informative for the task of classifying documents under class c_j whenever its presence (or absence) in a document d_i is a strong indicator of the fact that d_i belongs to c_j. For instance, `athlete` is an informative feature for class Sports (since it tends to occur more frequently in documents about Sports than in other documents), while `air` is

☆ This is a substantially revised and extended version of an extended abstract presented at the 2nd Italian Information Retrieval Workshop (IIR 2011). The order in which the authors are listed is purely alphabetical; each author has given an equally important contribution to this work.

* Corresponding author.
E-mail address: Fabrizio.Sebastiani@isti.cnr.it (F. Sebastiani).

[1] From now on we will use the terms "word", "feature", and "term", interchangeably, consistently with most IR literature.

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.02.010

not (since it tends to occur with approximately the same frequency in documents about Sports and in other documents).

In text classification, the most popular approach to feature selection is the *filter* approach (John et al., 1994). By and large, this consists of a greedy strategy in which a function f is applied to each feature $t_k \in W$ in order to compute a real-valued score $f(t_k)$ that represents its expected contribution to solving the classification task. Only the $|S|$ features with the highest f value are retained and will thus play a role in the training and classification phases, while the others are discarded from consideration.[2]

The most popular instances of function f above, such as pointwise mutual information, chi-square, odds ratio, information gain (also known as mutual information), bi-normal separation, and the like, are based on *binary* information indicating the presence/absence of the feature in each training document. For instance, in pointwise mutual information, defined as $PMI(t_k, c_j) = \log_2 \frac{P(t_k,c_j)}{P(t_k)P(c_j)}$, the value $P(t_k)$ is the probability that feature t_k occurs in a random training document ($P(c_j)$ is instead the probability that a random training document is a positive example of class c_j). As such, *PMI* and all the other above-mentioned functions do not exploit the information concerning *how many times* t_k occurs in a given training document; in other words, if d_i is a training example of c_j, whether it contains a single occurrence or multiple occurrences of t_k does not have an impact on $PMI(t_k,c_j)$.

This is counterintuitive, since the number of times a term occurs in a document (*term frequency*) is a rich source of information that should not be neglected, as witnessed from the fact that the history of term weighting in information retrieval (see e.g. Salton & Buckley, 1988; Robertson & Walker, 1994; Zobel & Moffat, 1998) has clearly shown the superiority of weighting approaches that make use of term frequency with respect to approaches that do not use it.

In this paper we propose a filter approach to feature selection for text classification which attempts to overcome this drawback. Rather than proposing new feature selection *techniques* alternative to those which only accommodate binary information (such as, e.g, *PMI*), our strategy is to stick to the latter ones but modify the input which is presented to them, so that they receive *term frequency information encoded as binary information*. The net effect will be that of allowing the use of all the standard feature selection functions based on binary information, while at the same time bringing to bear term frequency information. This has evident advantages, in the fact that the "classical" feature selection functions above are well-studied, have withstood the test of time, and have proven to be the best performers in all large-scale comparative text classification experimentation carried out to date (see e.g., Forman, 2003; Yang & Pedersen, 1997).

As the testbed for our approach we use "ordinal" text classification, a type of text classification (recently made popular by applications in customer relationship management, market research, and social Web mining) which is concerned with classes lying on an ordinal (i.e., totally ordered) rating scale; an example such set of classes may be $R = \langle \text{Disastrous} \prec \text{Bad} \prec \text{Fair} \prec \text{Good} \prec \text{Excellent} \rangle$ (as used, e.g., in rating products in customer reviews). We will test this approach in the context of (non-ordered) "single-label" classification (i.e., when exactly one class must be attached to a given document and the classes are not ordered) and "multi-label" classification (i.e., when zero, one, or more classes at the same time may be attached to a given document) in a future paper.

This paper is organized as follows. In Section 2 we present our μ-document-based approach to feature selection, and we briefly describe (in Section 2.2) the four feature selection methods for ordinal text classification that we will use as testbeds. Section 3 reports the results of experiments we have conducted using two SVM-based learning methods and two large datasets of product reviews. Section 4 concludes by pointing at avenues for future research.

2. Feature selection for OC based on training μ-documents

2.1. Encoding term frequency information as binary information

Let us fix some terminology. We take a document d_i to consist of a sequence $\theta_1\theta_2\cdots\theta_{length(d_i)}$ of *tokens*; each token θ_z is the occurrence of a given word, which we denote by $w(\theta_z)$. Given a set Tr of training documents, let us define $\Theta = \{\theta_z \in d_i | d_i \in Tr\}$ to be the set of all tokens contained in Tr, and $W = \{w(\theta_z) | \theta_z \in \Theta\}$ to be the set of unique words that occur at least once in at least one document of Tr.

As indicated in the introduction, our approach to feature selection for text classification does *not* consist in proposing new feature selection techniques alternative to the classic ones. It instead consists of modifying the input which is presented to standard techniques based on binary information, so that they receive term frequency information "disguised" as binary information.

Specifically, our approach consists of

1. breaking down (logically, and for the sole purpose of giving input to the feature selection function) each training document d_i into $length(d_i)$ training "micro-documents" (hereafter: μ-documents), each consisting of a single token and endowed with the same class label(s) of the original training document;
2. feeding the resulting training μ-documents (instead of the original documents) to the feature selection function of choice.

The net effect is that, if a training document d_i is a positive example of class c_j and contains $q > 1$ instances of term t_k, the feature selection function will receive as input q positive training examples of class c_j and containing t_k instead of a single one, thus *de facto* increasing the importance of t_k for class c_j. As a result, in the *PMI* formula of Section 1, $P(t_k)$ will no more denote the fraction $\frac{|\{d_i \in Tr | t_k = w(\theta_z), \theta_z \in d_i\}|}{|Tr|}$ of training documents that contain t_k, but the fraction of training *micro*-documents that contain t_k, which in turn corresponds to the fraction $\frac{|\{\theta_z | w(\theta_z) = t_k\}|}{|\Theta|}$ of all tokens in Θ that are instances of t_k. Similarly, $P(c_j)$ will now denote the fraction of μ-documents that are positive instances of c_j, that corresponds to the fraction $\frac{|\{\theta_z \in c_j\}|}{|\Theta|}$ of tokens that are positive instances of c_j.

The consequence of this move is that feature selection functions are now influenced not by the number of documents in c_j that contain an instance of t_k, but by the number of instances of t_k that are contained in documents in c_j. In practice, this has the effect of allowing term frequency to have an impact on feature selection, while at the same time allowing (thanks to the fact that tokens are now considered documents in their own right) the use of the "classic" feature selection functions.

Example 1. Assume that $Tr = \{d_1, d_2, d_3, d_4, d_5\}$; assume that d_1 and d_2 are positive examples of class c_j while d_3, d_4 and d_5 are negative examples of c_j; assume that term t_k occurs 4 times in each of d_1 and

[2] In this paper we do not consider so-called "wrapper" approaches to feature selection since they suffer from computational problems due to they combinatorial nature, and are thus used in practice only when the dimensionality of the original feature space is small, which is never the case for text classification tasks. It is thus completely inadequate for *text* learning tasks, in which the dimensionality of the original feature space is typically $O(10^5)$ or more (Interestingly, the literature on FS for metric regression seems to have mostly, if not only, investigated "wrapper" approaches (Miller, 2002).); the same happens for bioinformatics applications such as, e.g., gene selection for patient classification (Guyon, Weston, Barnhill, & Vapnik, 2002).

d_2, and only once in each of d_3, d_4 and d_5; and assume that each of d_1 and d_2 contains six other occurrences of words different from t_k and each of d_3, d_4 and d_5 contains nine of them.

We intuitively feel that t_k is a discriminator of some interest for c_j. Notwithstanding this, if we adopt the model based on "regular" documents, we have $PMI(t_k, c_j) = \log_2 \frac{P(t_k,c_j)}{P(t_k)P(c_j)} = \log_2 \frac{\frac{2}{5}}{1\times\frac{2}{5}} = 0$, i.e., t_k is considered utterly uninteresting as a discriminator for c_j. If we instead adopt the model based on μ-documents, we have $PMI(t_k, c_j) = \log_2 \frac{P(t_k,c_j)}{P(t_k)P(c_j)} = \log_2 \frac{\frac{8}{50}}{\frac{11}{50}\times\frac{20}{50}} = 0.036$, i.e., t_k is considered a discriminator of some interest for c_j. Other feature selection functions such as information gain, chi-square, and others, would behave similarly to *PMI*. □

The example above allows us to appreciate two reasons why the μ-documents approach is promising. The first reason is that, as we have extensively argued, multiple occurrences of the same term are now taken into account. This seems intuitively plausible: while a single occurrence of a word in a given document may be due to chance, it is much less likely that multiple occurrences of the same word in the document are also due to chance. The second reason is that the μ-documents approach intuitively appears more robust from a statistical point of view, since the counts that are fed into the feature selection functions are much higher. For instance, dataset TripAdvisor-15763 (see Table 1) contains only 10,508 "regular" training documents but 2,222,578 training μ-documents, a number 211 times bigger. This means that its 36,670 unique words will generate, on average, counts which are much higher, and the resulting scores will be much less affected by sparsity.

The move from training documents to training μ-documents is, as far as feature selection is concerned, akin to the move, in naïve Bayesian learners, from a multivariate Bernoulli event model (where documents are events) to a multinomial event model (where word occurrences are events). In the context of text classification this move was originally discussed in McCallum and Kamal (1998). However, in that case the authors reported that little difference in performance was found when selecting features via the former model rather than via the latter model (no actual effectiveness figures were given, though). Our work may be seen as exporting that idea outside the realm of naïve Bayesian learners, and outside the realm of single-label text classification, neither of which has been done before to the best of our knowledge.

It is very important to note that after the reduced set of features S has been identified via our feature selection mechanisms, classifier training proceeds as usual, i.e., by using "regular" training documents; that is, the training documents are broken up into μ-documents only logically, and only for the purpose of carrying out feature selection.

Note that switching from regular training documents to training micro-documents for feature selection purposes does not entail substantially higher costs from a computational point of view. In fact, from a practical point of view there is certainly no need (nor it makes any sense) to explicitly *generate* the μ-documents; as we have previously observed, the training documents are broken down into μ-documents only *logically*. The only thing that one needs to do in practice is feed the feature selection functions the term counts (i.e., the values of $P(t_k)$, $P(c_j)$ and $P(t_k, c_j)$) that would be obtained *if* the training documents were broken down into μ-documents.

Table 1
Main characteristics of the two datasets used in this paper; the four columns indicate the number of training documents, the number of test documents, the number of unique words, and the number of training μ-documents (i.e., the number of tokens), respectively.

Dataset	$\lvert Tr\rvert$	$\lvert Te\rvert$	$\lvert W\rvert$	$\lvert\Theta\rvert$
TripAdvisor-15763	10,508	5255	36,670	2,222,578
Amazon-83713	20,000	63,713	138,964	3,399,721

2.2. Feature selection methods for ordinal text classification

As indicated in the introduction, we here use ordinal text classification as a testbed of our idea. *Ordinal classification* (also known as *ordinal regression for text*) consists in estimating (from a training set Tr) a *target function* Φ: $D \rightarrow R$ which maps each document $d_i \in X$ into exactly one of an ordered sequence (that we here call *rankset*, or *rating scale*) $R = \langle r_1 \prec \cdots \prec r_n \rangle$ of *ranks* (aka "scores", or "labels", or "classes"). The result of the estimation is a function $\widehat{\Phi}$ called the *classifier*,[3] which we will evaluate on a test set Te. This problem is somehow intermediate between *single-label classification*, in which R is instead an unordered set, and *metric regression*, in which R is instead a continuous, totally ordered set (typically: the set $\mathbb{R}$ of the reals).

Our feature selection methods will typically consist of (a) attributing a score to each feature $t_k \in W$ by means of a function *Score* that measures the predicted utility of t_k for the classification process (the higher the value of *Score*, the higher the predicted utility), and, (b) given a predetermined reduction level ξ, selecting the $|S| = \xi \cdot |W|$ features based on their *Score*. The *Score* function will sometimes be rank-specific (i.e., its form will actually be $Score(t_k, r_j)$), while sometimes it will be global to the entire rankset (i.e., its form will actually be $Score(t_k)$); in the former case, a method for selecting a set of features global to the entire rankset from the rank-specific scores will also be needed.

We now briefly sketch the feature selection methods that we use here as testbeds for our idea. These are the *Var∗IDF*, *RR*(*Var∗IDF*), *RR*(*IGOR*) and *RR*(*AC∗IDF*) methods originally defined in Baccianella, Esuli, and Sebastiani (2010b) (an extended version of Baccianella, Esuli, & Sebastiani (2010a)). These functions represent the state of the art in feature selection for ordinal classification, since the experiments reported in Baccianella et al. (2010b) have shown that they substantively outperform the only two other such functions (*Var* – Shimada & Endo, 2008 and *PRP* – Mukras, Wiratunga, Lothian, Chakraborti, & Harper, 2007) ever discussed (to the best of our knowledge) in the ordinal regression literature.

For reasons of brevity we only describe *Var∗IDF*, *RR*(*Var∗IDF*), *RR*(*IGOR*) and *RR*(*AC∗IDF*) concisely; for more mathematical detail and for the intuitions that underlie them see Baccianella et al. (2010b). However, it is important to note that a detailed comprehension of them is not essential for the comprehension of this paper: for the purposes of our work they can essentially be seen as black boxes that, like the *PMI* function described in Section 1,

1. take as input a training set of documents labelled according to the rankset of choice;
2. return a *Score* for each term t_k of which there is an instance in Tr, obtained by analyzing the presence/absence of t_k in the positive and negative examples of c_j.

2.2.1. Var∗IDF

In the first method (*Var∗IDF*), $Score(t_k)$ is computed as

$$Score(t_k) = -(Var(t_k) + \epsilon) * (IDF(t_k))^a \quad (1)$$

where

[3] Consistently with most mathematical literature we use the caret symbol ($\wedge$) to indicate estimation.

- $Var(t_k)$ is the variance of the distribution across the ranks in R of the training documents containing t_k; only the fact that a training document contains or does not contain t_k is used;
- ϵ is a small positive constant whose purpose is to prevent the first factor in the multiplication from being equal to zero, which would reward those features that occur in a single document of Tr;
- $IDF(t_k) = \log_e \frac{|Tr|}{\#_{Tr}(t_k)}$ (where $\#_{Tr}(t_k)$ denotes the number of training documents that contain feature t_k) represents inverse document frequency;
- a is a nonnegative real-valued parameter (to be optimized on a validation set) that allows to fine-tune the relative contributions of $(Var(t_k) + \epsilon)$ and $IDF(t_k)$ to the product.

The $|S|$ features with the highest $Score(t_k)$ value are retained while the others are discarded.

2.2.2. RR(Var∗IDF)

The second method ($RR(Var{*}IDF)$) also uses Eq. (1) for computing $Score(t_k)$, but does not simplistically choose the $|S|$ top-scoring features. Instead, it provisionally assigns each feature t_k to the rank closest to the mean of the distribution across the ranks of the training documents containing it. Then, it performs a so-called "round robin" (RR) step, i.e., a step in which (i) for each rank $r_j \in R$ it sorts the features t_k assigned to r_j in descending order of $Score(t_k)$, and then (ii) it allows the n ranks $r_1, \ldots, r_n$ to take turns in picking features, one at a time, from the top of their rank-specific orderings, until $|S|$ features have been picked.

2.2.3. RR(IGOR)

The third method ($RR(IGOR)$ – where IGOR stands for "information gain for ordinal regression"), computes scores via a rank-specific function $Score(t_k, r_j)$. For each feature t_k and for each $j = 1, \ldots, (n-1)$ we define $c_j = r_1 \cup \cdots \cup r_j$ and $\bar{c}_j = r_{j+1} \cup \ldots \cup r_n$ and compute

$$Score(t_k, r_j) = IG(t_k, c_j) = \sum_{x \in \{c_j, \bar{c}_j\}} \sum_{y \in \{t_k, \bar{t}_k\}} P(x,y) \log_2 \frac{P(x,y)}{P(x)P(y)} \quad (2)$$

Here probabilities are interpreted in an event space of documents; this means that, e.g., $P(c_j)$ stands for the probability that a random document belongs to class c_j, and $P(\bar{t}_k)$ stands for the probability that a random document does not contain t_k. That is, we use the classic information gain function as if we had to perform feature selection for a binary classification task in which the two classes to separate are the union of the first j ranks and the union of the last $(n-j)$ ranks in the rankset.

We then (i) sort, for each of the ranks in $\{r_1, \ldots, r_{n-1}\}$, the $|W|$ features in decreasing order of their $Score(t_k, r_j)$ value, and (ii) carry out a round robin step as for method $RR(Var{*}IDF)$, until $|S|$ features have been picked.

2.2.4. RR(AC∗IDF)

The fourth and last method ($RR(AC{*}IDF)$ – where AC stands for "anticorrelation"), also computes for each feature t_k, via a rank-specific function $Score(t_k, r_j)$, scores for each of the ranks $r_j \in R$ via the function

$$Score(t_k, r_j) = -\left(\frac{\sum_{\{d_i \in Tr | t_k \in d_i\}} E(\widetilde{\Phi}_j, d_i)}{|\{d_i \in Tr | t_k \in d_i\}|} + \epsilon \right) * (IDF(t_k))^a \quad (3)$$

where $\widetilde{\Phi}_j$ is the "trivial" classifier that assigns all documents to the same rank r_j, $E(\widetilde{\Phi}_j, d_i)$ is an error measure (here taken to be $|\widetilde{\Phi}(d_i) - \Phi(d_i)|$, i.e., the absolute distance between the rank predicted by $\widetilde{\Phi}_j$ and the true rank), and IDF, ϵ and a are as in Eq. (1). We then (i) sort, for each of the n ranks $r_j \in R$, the $|W|$ features in decreasing order of their $Score(t_k, c_j)$ value, and (ii) carry out a round robin step as in the two previous methods, until $|S|$ features have been picked.

Table 2
Main characteristics of the two datasets used in this paper; the five columns indicate, for each rank, the fraction of documents that belong to the rank.

Dataset	1 Star (%)	2 Stars (%)	3 Stars (%)	4 Stars (%)	5 Stars (%)
TripAdvisor-15763	3.9	7.2	9.4	34.5	45.0
Amazon-83713	16.2	7.9	9.1	23.2	43.6

2.3. Discussion

As repeatedly noted before, most popular feature selection functions from the text classification literature only use information indicating the presence/absence of feature t_k in training document d_i, and do not use information on the number of times t_k occurs in d_i. This is also true of the four methods we have presented in Section 2.2. In fact:

- In the $Var{*}IDF$ and $RR(Var{*}IDF)$ methods, $Var(t_k)$ is the variance of the distribution of the training documents containing t_k; that is, only the fact that a training document contains or does not contain t_k is used.
- Concerning $RR(IGOR)$, in Eq. (2) the quantity $P(t)$ is the probability that a random training document contains t at all; the number of times t occurs in the document has no impact.
- In the $RR(AC{*}IDF)$ method, the first factor of Eq. (3) depends, both at the numerator and at the denominator, on the training documents that contain t_k at all; the number of times t_k is contained in them has no impact.

3. Experiments

3.1. Experimental setting

3.1.1. The datasets

We have tested the proposed method on two different datasets,[4] whose characteristics are concisely reported in Tables 1 and 2.

The first is the TripAdvisor-15763 dataset first used in Baccianella, Esuli, and Sebastiani (2009b) and consisting of 15,763 hotel reviews from the TripAdvisor Web site. We use the same split between training and test documents as used in Baccianella et al. (2009b), resulting in 10,508 documents used for training and 5255 for test; the training set contains 36,670 unique words.

The second dataset is the Amazon-83713 dataset first used in Baccianella et al. (2010b) and consisting of 83,713 home electronics product reviews from the Amazon Web site. Amazon-83713 is actually a small subset of the Amazon dataset,[5] consisting of more than 5 million reviews, originally built by Jindal and Liu for spam review detection purposes (Jindal et al., 2007), and contains all the reviews in the sections `MP3`, `USB`, `GPS`, `Wireless 802.11`, `Digital Camera`, and `Mobile Phone`. We use the same split between training and test documents as in Baccianella et al. (2010b), resulting in 20,000 documents used for training and 63,713 for test; the training set contains 138,964 unique words. To the best of our knowledge, Amazon-83713 is still the largest dataset ever used in the literature on ordinal text classification.

[4] Both datasets are available for download from http://hlt.isti.cnr.it/reviewdata/.
[5] http://www.cs.uic.edu/liub/FBS/sentiment-analysis.html.

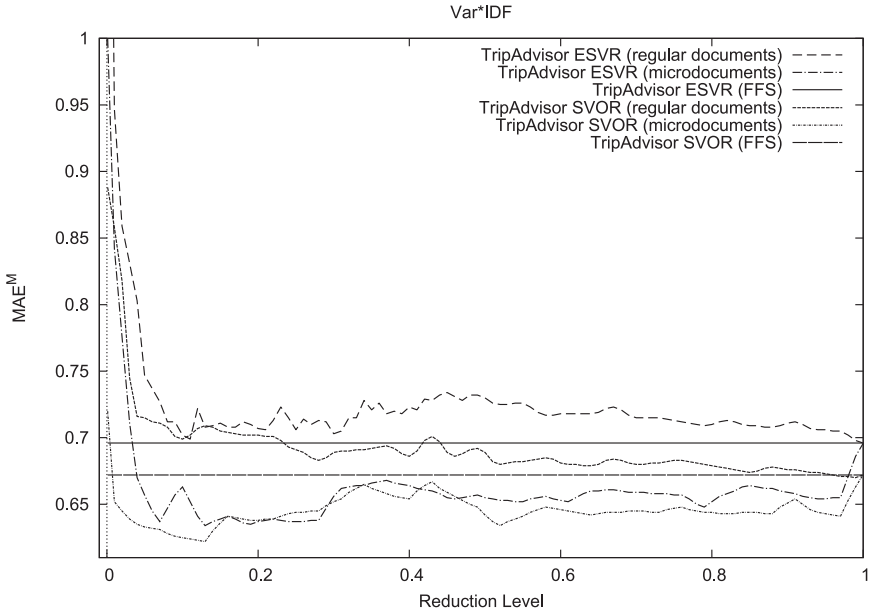

Fig. 1. Results obtained with the two variants (based on "regular" documents and on μ-documents, respectively) of the *Var∗IDF* feature selection function on the TripAdvisor-15763 dataset with the ϵ-SVR and SVORIM learners. Results are evaluated with MAE^M; lower values are better. "FFS" refers to the use of the full feature set (i.e., $\xi = 1$).

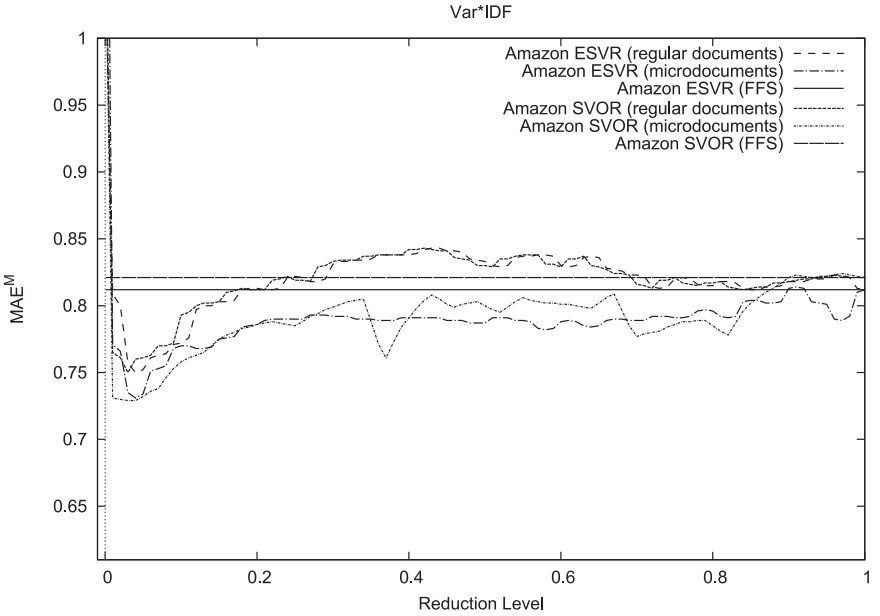

Fig. 2. Same as Fig. 1 but with Amazon-83713 in place of TripAdvisor-15763.

Both datasets consist of reviews scored on a rating scale from 1 Star to 5 Stars; both datasets are highly imbalanced (see Table 2), with positive and very positive reviews by far outnumbering negative and very negative reviews.

3.1.2. Evaluation measures

As our main evaluation measure we use the *macroaveraged mean absolute error* (MAE^M) measure proposed in Baccianella, Esuli, and Sebastiani (2009a), and defined as

$$MAE^M(\widehat{\Phi}, Te) = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{|Te_j|}\sum_{d_i \in Te_j}|\widehat{\Phi}(d_i) - \Phi(d_i)| \quad (4)$$

where Te_j denotes the set of test documents whose true rank is r_j and the "M" superscript indicates "macroaveraging". As argued in Baccianella et al. (2009a), the advantage of MAE^M over "standard" mean absolute error (defined as

$$MAE^{\mu}(\widehat{\Phi}, Te) = \frac{1}{|Te|}\sum_{d_i \in Te}|\widehat{\Phi}(d_i) - \Phi(d_i)| \quad (5)$$

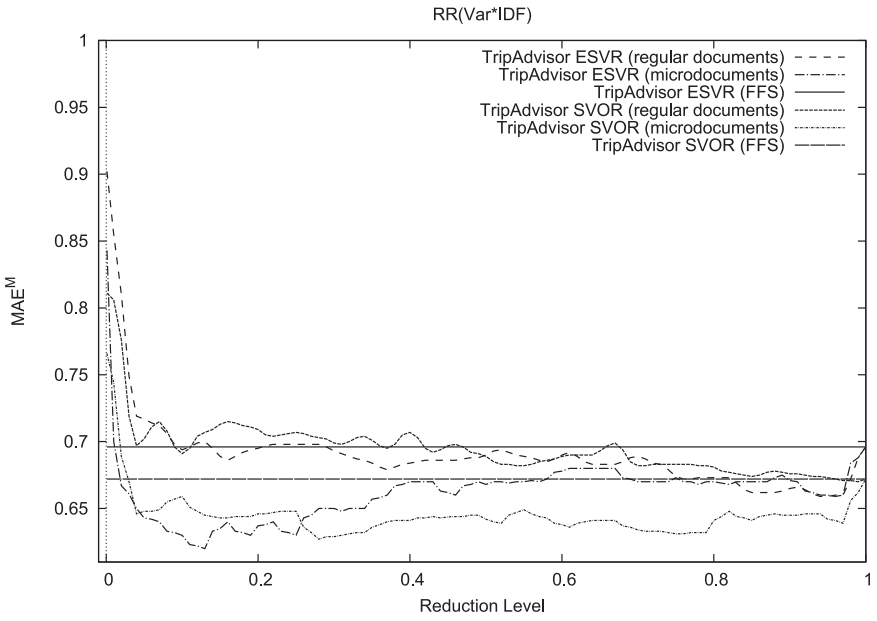

Fig. 3. Same as Fig. 1 but with *RR*(*Var∗IDF*) in place of *Var∗IDF*.

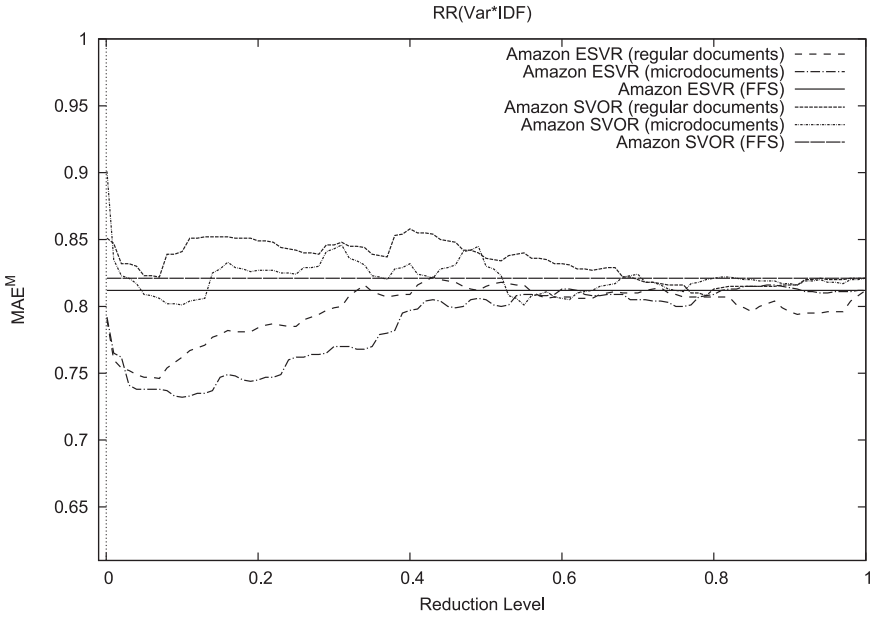

Fig. 4. Same as Fig. 3 but with Amazon-83713 in place of TripAdvisor-15763.

where the "μ" superscript stands for "microaveraging") is that it is robust to rank imbalance (which is useful, given the above-mentioned imbalanced nature of our datasets) while coinciding with MAE^{μ} on perfectly balanced datasets (i.e., datasets with exactly the same number of test documents for each rank).

3.1.3. Learning algorithms

We have tested our methods with two different SVM-based learning algorithms for ordinal regression: ϵ-SVR (Drucker, Burges, Kaufman, Smola, & Vapnik, 1997), originally devised for linear regression and which we have adapted to solve ordinal regression problems, and SVOR (Chu & Keerthi, 2007), which was specifically devised for solving ordinal regression.

ϵ-support vector regression (ϵ-SVR) is the original formulation of support vector (metric) regression as proposed in Drucker et al. (1997); we have used the implementation from the freely available LibSvm library.[6] ϵ-SVR can be adapted to the case of *ordinal* regression by (a) mapping the rankset onto a set of consecutive natural numbers (in our case we have simply mapped the sequence [1 Star,...,5 Stars] onto the sequence [1,...,5]), and (b) rounding the

[6] http://www.csie.ntu.edu.tw/cjlin/libsvm/.

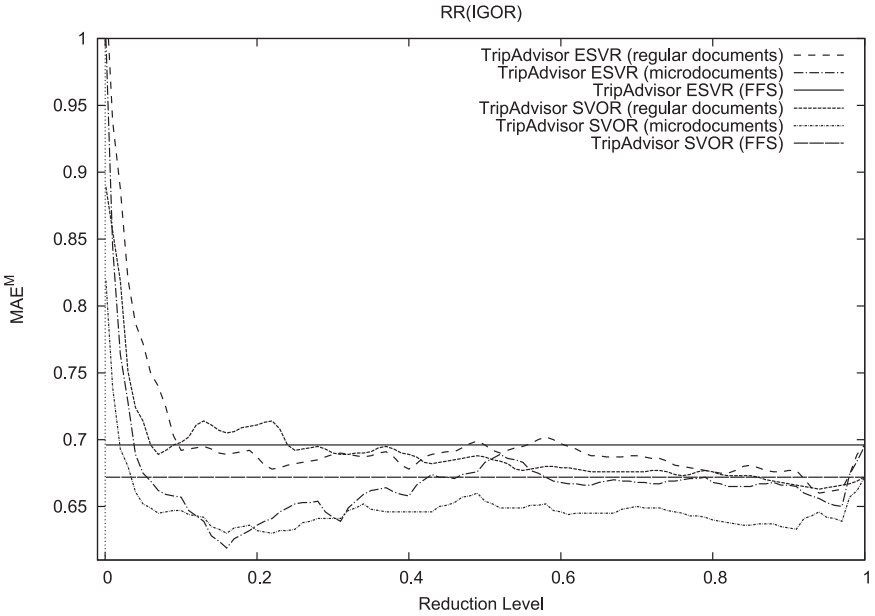

Fig. 5. Same as Fig. 1 but with *RR*(*IGOR*) in place of *Var∗IDF*.

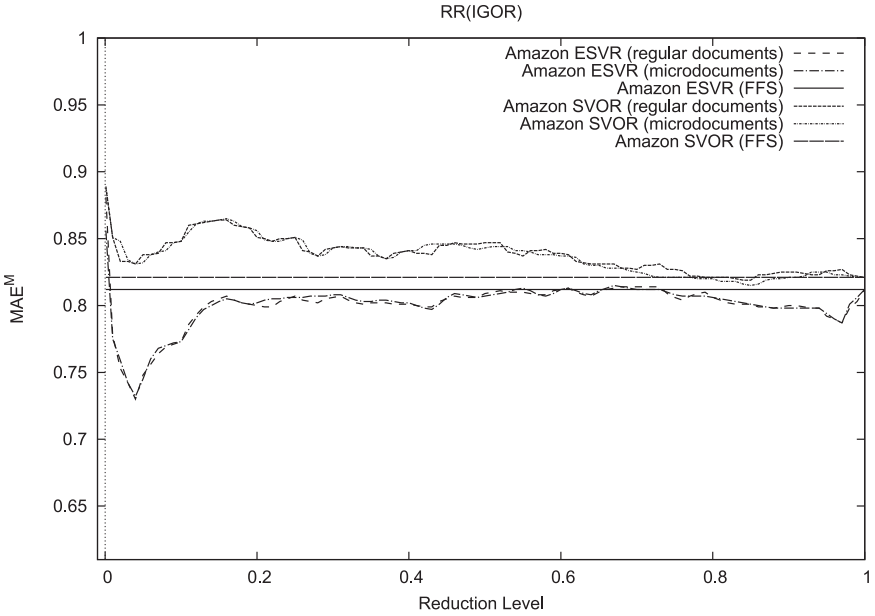

Fig. 6. Same as Fig. 5 but with Amazon-83713 in place of TripAdvisor-15763.

real-valued output of the classifier to the nearest natural number in the sequence.

SVOR (Chu & Keerthi, 2007) consists instead of a newer algorithm that tackles the ordinal regression problem without using any *a priori* information on the ranks, and by finding $n-1$ thresholds that divide the real-valued line into n consecutive intervals corresponding to the n ordered ranks. The authors propose two different variants: the first (nicknamed SVOREX, for "Support Vector Ordinal Regression with EXplicit constraints") takes into account only the training examples of adjacent ranks in order to determine the thresholds, while the second (SVORIM, for "Support Vector Ordinal Regression with IMplicit constraints") determines each threshold by using all the training examples from all of the ranks. Given that the authors have experimentally shown SVORIM to outperform SVOREX, the former (in the implementation[7] available from the authors of Chu & Keerthi (2007)) is the variant we have adopted for our experiments.

Both learning algorithms use the sequential minimal optimization algorithm for SVMs (Platt, 1999), and both map the solution

[7] http://www.gatsby.ucl.ac.uk/chuwei/svor.htm.

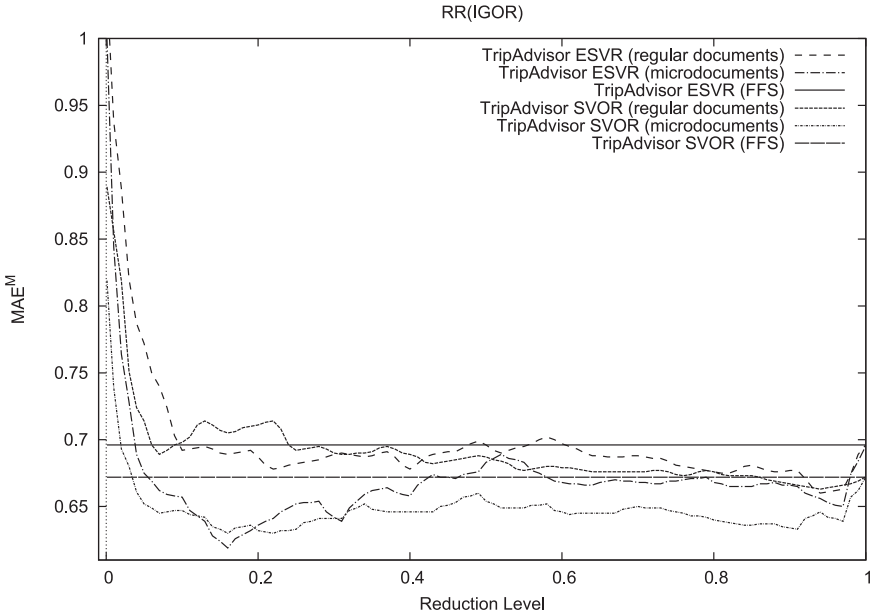

Fig. 7. Same as Fig. 1 but with *RR*(*AC*∗*IDF*) in place of *Var*∗*IDF*.

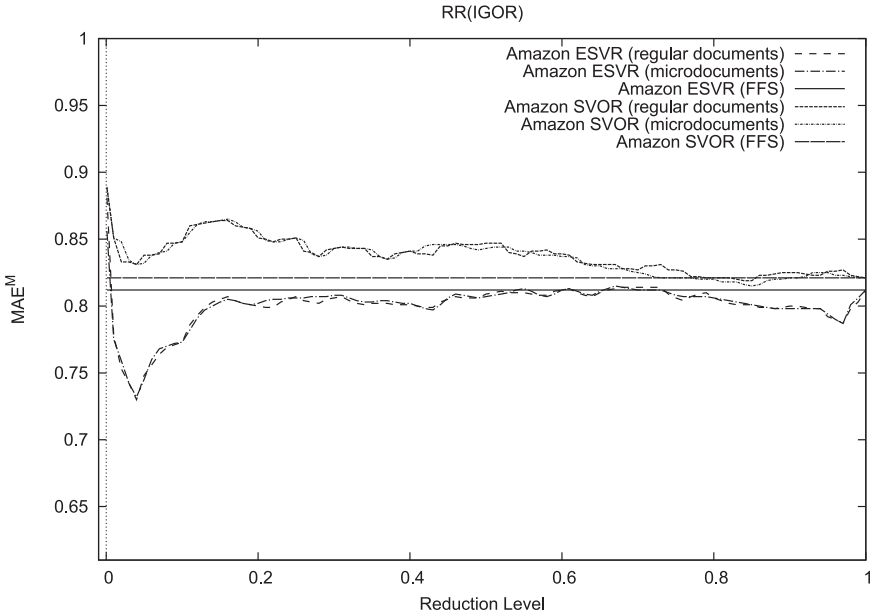

Fig. 8. Same as Fig. 7 but with Amazon-83713 in place of TripAdvisor-15763.

onto the real-valued line. The main difference between them is the use of *a priori* information. In fact, when using ϵ-SVR the user needs to explicitly specify a mapping of the rankset onto a sequence of natural numbers and to set the thresholds in-between these latter, while SVOR automatically derives all the needed information from the training set.

As the baselines against which to test our μ-documents-based approach we have used the results we have obtained in Baccianella et al. (2010b) (on the same datasets and with the same learning algorithms) with the versions based on "regular" training documents of the same feature selection functions.

3.1.4. Experimental protocol

The experimental protocol essentially conforms to that of Baccianella et al. (2010b). As a vectorial representation, after stop word removal (and no stemming) we have used standard bag-of-words with cosine-normalized *tfidf* weighting. We have run all our experiments for all the 100 reduction levels $\xi \in \{0.001, 0.01, 0.02, 0.03, \ldots, 0.99\}$. This results in a massive experimentation effort, consisting of 2 datasets $\times$ 2 learners $\times$ 100 reduction factors $\times$ 4 feature selection functions = 1600 train-and-test experiments, which add up to the other 1600 whose results we use as baseline and that had been already presented in Baccianella et al. (2010b).

We have set

- the γ and C parameters of both ϵ-SVR and SVOR, and
- the a parameter for the *Var∗IDF*, *RR*(*Var∗IDF*) and *RR*(*AC∗IDF*) methods

to the optimal values that we had obtained in the experiments of Baccianella et al. (2010b). This means that the parameters are optimal for the baselines but not necessarily for the methods proposed here, which lends even higher value to the results obtained by these latter.

For the *Var∗IDF*, *RR*(*Var∗IDF*) and *RR*(*AC∗IDF*) methods we have set the smoothing parameter ϵ to 0.1, i.e., to the same value we had used in the experiments of Baccianella et al. (2010b), so as to allow perfect comparability between the current experiments and the baseline experiments. For *RR*(*AC∗IDF*), the E error measure was taken to be $|\widehat{\Phi}(d_i) - \Phi(d_i)|$ (i.e., absolute error), given that it is the document-level analogue of our chosen measure (MAE^M).

3.2. Results

The results of our experiments are displayed in Figs. 1–8. In each such figure the effectiveness (measured via MAE^M) of one of our four feature selection functions is plotted as a function of the tested reduction level; in each figure four curves are reported, deriving from the choice of two learners and two different interpretations (based on "regular" documents – indicated by solid lines – or on μ-documents – indicated by dotted lines) of the same function. The horizontal lines indicate the effectiveness obtained by using the full feature set ($\xi = 1$, indicated as "FFS").

The main observation to be made from these plots is that the use of training μ-documents substantially enhances the accuracy of ordinal text classification, since it is practically always the case that the MAE^M values of the μ-document-based versions are better than the corresponding values of the classic, "regular document"-based versions, irrespective of feature selection function, dataset, and learner.

A second insight that can be obtained by analyzing these plots is that the use of feature selection functions based on μ-documents allows to obtain substantially smaller levels of error with reduced feature sets (i.e., with values of $\xi < 1$) than with the full feature set, which rarely happens (as evident from the plots) with the standard versions of the same functions. That this is a strikingly successful aspect of our method can be seen by considering the fact that these results have been obtained with learning algorithms in the support vector machine family, which have consistently been shown to be robust to very large dimensionalities of the feature space (Joachims, 1998; Taira & Haruno, 1999). In other words, SVMs are probably the toughest benchmark for a feature selection algorithm, and the improvements shown by our methods with respect to using the full feature set speak very much in favour of them.

Table 3 reports MAE^M values as averaged, for a given combination of dataset and learner, across the 100 values of ξ; these values show an average error reduction ranging from 2.48% (SVOR on TripAdvisor-15763) to 6.29% (ϵ-SVR on Amazon-83713), with even higher error reductions obtained for specific feature selection functions. A further interesting observation that this table allows to draw is that the improvements brought about by the μ-documents technique are much higher for ϵ-SVR than for SVOR; the fact that ϵ-SVR is practically always a better performer than SVOR lends thus to the μ-documents technique even higher value.

Table 3
Mean values of MAE^M computed across the 100 different values for ξ; Δ indicates the relative reduction in average MAE^M obtained by replacing regular training documents (RDs) with μ-documents (μDs).

	ϵ-SVR			SVOR		
	RDs	μDs	Δ (%)	RDs	μDs	Δ (%)
TripAdvisor-15763						
Var∗IDF	0.722	0.658	(−8.84)	0.818	0.787	(−3.82)
RR(*Var∗IDF*)	0.688	0.660	(−4.10)	0.797	0.787	(−1.36)
RR(*IGOR*)	0.695	0.665	(−4.25)	0.800	0.800	(−0.00)
RR(*AC∗IDF*)	0.680	0.666	(−2.09)	0.818	0.780	(−4.71)
Average	0.696	0.662	(−4.87)	0.808	0.788	(−2.48)
Amazon-83713						
Var∗IDF	0.691	0.645	(−6.69)	0.818	0.790	(−3.51)
RR(*Var∗IDF*)	0.694	0.644	(−7.26)	0.833	0.821	(−1.52)
RR(*IGOR*)	0.689	0.646	(−6.23)	0.838	0.837	(−0.12)
RR(*AC∗IDF*)	0.697	0.662	(−4.99)	0.837	0.787	(−5.92)
Average	0.693	0.649	(−6.29)	0.831	0.808	(−2.77)

4. Conclusions and further research

We have presented a method for performing feature selection in text classification contexts that allows classic feature selection methods based on binary information to be sensitive to term frequency information, i.e., to multiple occurrences of the same term in a given training document. We have obtained this by encoding term frequency information as binary information; more specifically, we have obtained this by (logically) breaking down each training document d_i into $length(d_i)$ "micro-documents" (i.e., documents consisting of a single word occurrence) labelled with the same class label(s) as the original document. As a testbed for this method we have used ordinal text classification. We have presented a large-scale experimentation in which we have tested this method with two datasets of product reviews, four previously presented feature selection functions, and two learning algorithms for ordinal regression. The results have shown substantial accuracy improvements for all combinations of dataset, feature selection function, and learning algorithm.

One of the problems with the μ-documents-based method we have proposed is that, by transforming each word occurrence into a μ-document and then considering it as a training example for the purposes of feature selection, it *de facto* enforces a notion of term frequency in which this latter grows *linearly* with the number $\#(t_k, d_i)$ of occurrences of feature t_k in document d_i, i.e., $tf(t_k, d_i) = \#(t_k, d_i)$. It is instead well-known that the best-performing variants of the *tf* function are the ones in which $tf(t_k, d_i)$ grows *sublinearly* with $\#(t_k, d_i)$ (Salton & Buckley, 1988; Zobel & Moffat, 1998); a simple example is the well-known form

$$tf(t_k, d_i) = \begin{cases} 1 + \log \#(t_k, d_i) & \text{if } \#(t_k, d_i) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

In order to address this shortcoming we plan to experiment a variant of this approach in which, after breaking a training document down into μ-documents, only a fraction of these μ-documents are retained for feature selection. Given a nonzero number $\#(t_k, d_i)$ of occurrences of feature t_k in training document d_i, this approach would consist of retaining, instead of $\#(t_k, d_i)$ μ-documents consisting of feature t_k and derived from breaking up d_i, only $1 + \lfloor \log \#(t_k, d_i) \rfloor$ (or $1 + \lceil \log \#(t_k, d_i) \rceil$) of them. This would *de facto* enforce the notion of term frequency formalized by Eq. (6).

References

Baccianella, S., Esuli, A., & Sebastiani, F. (2009). Evaluation measures for ordinal text classification. In *Proceedings of the ninth IEEE international conference on intelligent systems design and applications (ISDA 2009)* (pp. 283–287). Pisa, IT.

Baccianella, S., Esuli, A., & Sebastiani, F. (2009). Multi-facet rating of product reviews. In *Proceedings of the 31st European conference on information retrieval (ECIR 2009)* (pp. 461–472). Toulouse, FR.

Baccianella, S., Esuli, A., & Sebastiani, F. (2010). Feature selection for ordinal regression. In *Proceedings of the 25th ACM symposium on applied computing (SAC 2010)* (pp. 1748–1754). Sierre, CH.

Baccianella, S., Esuli, A., & Sebastiani, F. (2010). Feature selection for ordinal text classification. Technical report 2010-TR-014, Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, IT.

Chu, W., & Keerthi, S. (2007). Support vector ordinal regression. *Neural Computation, 19*(3), 145–152.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A., & Vapnik, V. (1997). Support vector regression machines. In *Proceedings of the ninth conference on neural information processing systems (NIPS 1996)* (pp. 155–161). Denver, US.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research, 3*, 1289–1305.

Forman, G. (2007). Feature selection for text classification. In L. Huan & M. Hiroshi (Eds.), *Computational methods of feature selection* (pp. 257–276). London, UK: CRC Press/Taylor and Francis Group.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning, 46*(1/3), 389–422.

Huan, L., & Hiroshi, M. (Eds.). (2007). *Computational methods of feature selection*. London, UK: CRC Press/Taylor and Francis Group.

Isabelle, G., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*, 1157–1182.

Jindal, N., & Liu, B. (2007). Review spam detection. In *Proceedings of the 16th international conference on the world wide web (WWW 2007)* (pp. 1189–1190). Banff, CA.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European conference on machine learning (ECML 1998)* (pp. 137–142). Chemnitz, DE.

John, G. H., Kohavi, R. & K. Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of the 11th international conference on machine learning (ICML 1994)* (pp. 121–129). New Brunswick, US.

Maghsoodi, N., & Homayounpour, M. M. (2011). Improving Farsi multiclass text classification using a thesaurus and two-stage feature selection. *Journal of the American Society for Information Science and Technology, 62*(10), 2055–2066.

McCallum, A. K. & Kamal, N. (1998). A comparison of event models for naive Bayes text classification. In *Proceedings of the AAAI workshop on learning for text categorization* (pp. 41–48). Madison, US.

Mengle, S. S. R., & Goharian, N. (2009). Ambiguity measure feature-selection algorithm. *Journal of the American Society for Information Science and Technology, 60*(5), 1037–1050.

Miller, Alan (2002). *Subset selection in regression* (2nd ed.). London, UK: Chapman and Hall.

Mukras, R., Wiratunga, N., Lothian, R., Chakraborti, S., & Harper, D. (2007). Information gain feature selection for ordinal text classification using probability re-distribution. In *Proceedings of the IJCAI 2007 workshop on text mining and link analysis*. Hyderabad, IN.

Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, & A. J. Smola (Eds.), *Advances in kernel methods: Support vector learning* (pp. 185–208). Cambridge, US: MIT Press.

Robertson, S. & Walker, S. (1994). Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th ACM international conference on research and development in information Retrieval (SIGIR 1994)* (pp. 232–241). Dublin, IE.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management, 24*(5), 513–523.

Shimada, K., & Endo, T. (2008). Seeing several stars: A rating inference task for a document containing several evaluation criteria. In *Proceedings of the 12th Pacific-Asia conference on knowledge discovery and data mining (PAKDD 2008)* (pp. 1006–1014). Osaka, JP.

Taira, H., & Haruno, M. (1999). Feature selection in SVM text categorization. In *Proceedings of the 16th conference of the American association for artificial intelligence (AAAI 1999)* (pp. 480—486). Orlando, US.

Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *Proceedings of the 14th international conference on machine learning (ICML 1997)* (pp. 412–420). Nashville, US.

Zobel, J., & Moffat, A. (1998). Exploring the similarity space. *SIGIR Forum, 32*(1), 18–34.