Similarity-Based Synthetic Document Representations for Meta-Feature Generation in Text Classification

Sergio Canuto
DCC–UFMG
Belo Horizonte, Brazil
sergiodaniel@dcc.ufmg.br

Thiago Salles
DCC–UFMG
Belo Horizonte, Brazil
tsalles@dcc.ufmg.br

Thierson C. Rosa
INF–UFG
Goiânia, Brazil
thierson@inf.ufg.br

Marcos A. Gonçalves
DCC–UFMG
Belo Horizonte, Brazil
mgoncalv@dcc.ufmg.br

ABSTRACT

We propose new solutions that enhance and extend the already very successful application of meta-features to text classification. Our newly proposed meta-features are capable of: (1) improving the correlation of small pieces of evidence shared by neighbors with labeled categories by means of synthetic document representations and (local and global) hyperplane distances; and (2) estimating the level of error introduced by these newly proposed and the existing meta-features in the literature, specially for hard-to-classify regions of the feature space. Our experiments with large and representative number of datasets show that our new solutions produce the best results in all tested scenarios, achieving gains of up to 12% over the strongest meta-feature proposal of the literature.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning approaches**;

ACM Reference Format:
Sergio Canuto, Thiago Salles, Thierson C. Rosa, and Marcos A. Gonçalves . 2019. Similarity-Based Synthetic Document Representations for Meta-Feature Generation in Text Classification. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3331184.3331239

1 INTRODUCTION

Automatic Text Classification is a primary application of supervised learning, involving the automatic assignment of text documents to pre-defined classes. Recent developments in this area exploit data engineering solutions in the form of distance-based meta-features which can replace or augment the original set of (bag-of-words-based) features [3, 4, 10, 16, 19]. Such manually designed meta-features are extracted from original features (text) and are able to capture information from distance relationships among documents (by considering the location of training documents in the original feature space). The main assumption is that close documents tend to belong to the same class. These meta-features can capture insightful new information about the unknown underlying data distribution that relates the observed patterns with the associated category. Previous work reported successful results on improving classification effectiveness by using a compact meta-feature representation extracted from distance scores (e.g., distance between a document and its neighbors from each category [6, 19], neighborhood statistics [3, 4], and distances between documents and category centroids [14]).

Despite the previous success, the underlying distance relationships considered by previous work rely on traditional distance measures among documents. These distances aim at summarizing discriminative evidence based on simple manipulations of term weights (such as TF-IDF), which might thwart the importance of relevant discriminative terms in the similarity computation. Also, distance measures such as Cosine, Euclidean and Manhattan are not designed to capture whether two documents belong to the same class and thus do not directly associate similarity with class information.

In this paper, we tackle these limitations by proposing two types of distance based meta-features that correlate a set of similarity evidences of a pair of documents with the likelihood of these documents belonging to the same class. Those meta-features are as follows:

Distance-based meta-features from Synthetic Document Representations (SDRs). The first type of meta-features we propose uses SDRs built from similarity evidence (e.g., common words among documents and similarity measures) found on nearby documents to correlate pair of documents with classes. As illustrated in Figure 1, the common words between two documents and the similarity scores between them provide features for the resulting SDR.

SDRs and the class labels of the pairs of documents originating them are used to generate a training collection. A SDR in this collection is labeled as positive if the the pair belongs to the same class and as negative, otherwise.

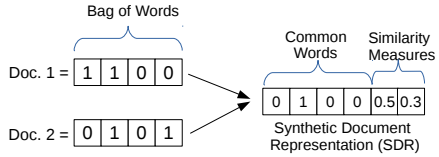

Figure 1: SDR built from similarity evidence.

A predictor (in our case, an SVM classifier) is learned from this "synthetic collection", producing a hyperplane able to separate positive from negative SDRs. In other words, an effective predictor provides high scores (i.e., high distances

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331239

from the hyperplane) when there is compelling similarity evidence to assert that two original documents in the pair belong to the same class.

Once the predictor is learned, we can generate our first type of meta-features for a given document as follows. Given a target document t on the original collection, we first obtain the neighbors of t belonging to the original training set (i.e., neighbor documents whose classes are known). For each neighbor n, we obtain a SDR formed by n and t. Next, we compute the distance between each resulting SDR and the hyperplane previously obtained. These distances correspond to our first type of meta-features for the target document.

The left part of Figure 2 illustrates the generation of this type of meta-feature for a target document represented in the figure by a circle with a "?" inside. The circles connected to the target by a line represent the neighbors of ? in the original training set. There are four neighbors in this example. Each pair $(?, neighbor)$ produces a single SDR using similarity evidence shared by the two documents in the pair. The SDR production is emphasized by the circled pair pointing (with a dashed arrow) to a SDR in the right side of the figure. The four distances between the hyperplane and the SDRs (represented as double-arrow lines in the right part of the figure) correspond to the proposed meta-features for document "?". These hyperplane distances directly correlate with the discriminative power of the similarity evidence contained in each SDR. The higher the distances, the stronger the evidence. In sum, instead of relying on the cosine similarity score of pairs $(?, neighbor)$ to produce features as in [4] (i.e., using only the left part of Figure 2), we propose to use the hyperplane distances corresponding to such pairs, which can better assess the discriminative power of the similarity evidence.

Error rate based meta-features. An important aspect of the process to construct the proposed meta-features is that it allows us to identify hard-to-classify examples. Given a target document, it is possible to identify if it is hard-to-classify by analyzing both the proportion of errors in the predictions for SDRs formed from the target document and the differences in the distances among these SDRs and the hyperplane. For example, consider the four SDRs formed from the target "?" and its neighbors in Figure 2. The one represented by a green-border square positioned above the hyperplane was wrongly classified by the predictor. Since this SDR was formed by similarity evidence between the target document and a neighbor with label "1", the predictor was "fooled" by the similarity evidence in the SDR. Thus, the proportion of SDRs wrongly classified by the predictor provides evidence regarding hard-to-classify documents. Accordingly, such proportion (of incorrectly classified SDRs) corresponds to the second type of proposed meta-features.

Notice that this second type of meta-feature works as an evaluation of the quality of the first type of meta-feature. Whenever the former meta-features indicate that a document is difficult to classify, supplementary information about the neighborhood of a target document is needed. In this case, we propose to combine our proposed meta-features based on SDRs with an extended version of the meta-features that presented the best results in [4].

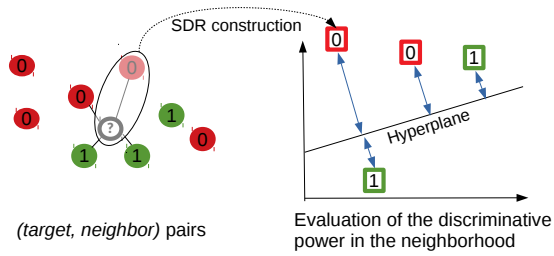

Figure 2: Evaluating the discriminative power of similarity evidence among the neighbors of a target document with a hyperplane as a predictor. Hyperplane distances provide new meta-features to represent the target document. Circles and squares indicate original and SDRs, respectively. Labels 0 and 1 indicate their associated category.

Different sets of the proposed meta-features can be obtained if we use different training sets or different algorithms to find the separating hyperplane for the SDRs. Particularly, we construct two different kinds of hyperplanes to evaluate SDRs by using different training sets. The first kind uses SDRs produced with all training documents as training examples to produce a hyperplane. In this scenario, the global similarity information related to all training documents is explored when learning the hyperplane, and the distances between a SDR and the hyperplane reflect the use of all training information. The second kind of hyperplane is inspired on the SVM-kNN method [20]. We build local hyperplanes using only the nearest SDRs of a target document as training examples. In this scenario, the hyperplane construction ignores potentially unrelated documents beyond the neighborhood of a document by locally adjusting the capacity of the SVM hyperplane to the properties of the training set in each localized area of the input space of SDRs.

Thus, we propose a meta-feature space that exploits not only distances from different hyperplanes, but also the identification of hard-to-classify examples and other statistics to replace (or extend) the original (bag-of-words) features of documents. Our experimental results in a large and heterogeneous set of datasets show significant improvements (up to 12%) of our proposal over a strong baseline constituted of the best literature meta-feature groups specially selected for each of the considered datasets. Such improvements helped our proposal to achieve the best results in *all tested* scenarios.

In sum, the main contributions of this paper are: (i) the design and evaluation of new meta-features based on SDRs and distances to hyperplanes especially designed to learn and evaluate discriminative similarity evidence; (ii) a new set of meta-features for estimating the level of error introduced by the newly proposed and the existing meta-features, specially for hard-to-classify regions of the feature space and (iii) an analysis of the effects of different groups of meta-features on classification effectiveness.

2 RELATED WORK

Several meta-features have been proposed to improve the effectiveness of machine learning methods. They exploit clustering methods [8, 9, 16], neighborhood of documents [1–4, 6, 19] or category centroids [14].

The use of clustering [8, 10, 16] to generate meta-features related to each cluster is among the earliest strategies to generate distance-based meta-features. Particularly, such strategies use the idea of aggregating the information of similar documents on clusters using both labeled and unlabeled data [8, 10, 16]. Such clusters produce meta-features that indicate the similarity of each example to potentially informative groups of documents. In [16] the largest n clusters are chosen as the most informative ones. Each cluster c contributes with a set of meta-features such as an indicator whether c is the closest of the n clusters to the example or not, the similarity of the example to the cluster's centroid, among others. In [8, 10] the number of clusters is chosen to be equal to the predefined number of classes and each cluster corresponds to an additional meta-feature.

Recently, several works [1–4, 6, 14, 19] have proposed to use similarities to category centroids or similarities between documents and its neighbors as their main source of information to generate meta-features. They differ from the previously described meta-features derived from clusters because they partition the training set for each class and exploit statistics from each training subset. [6] reported good results by designing meta-features that make a combined use of local information (through similarity scores among neighbors) and global information (through similarity to centroids) in the training set. This work was extended by the same authors in [19] by leveraging successful learning-to-rank retrieval algorithms over the meta-feature space for the multi-label classification problem. Latter, [2] proposed an efficient implementation with massively parallel manycore GPU architectures for the meta-features proposed in [6, 19].

An alternative strategy to efficiently generate meta-features for high dimensional and sparse data was provided in [14] to exploit a compact meta-feature space derived only from category centroids. The use of these meta-features brings benefits for both efficiency and classification effectiveness, especially for highly unbalanced datasets.

All the previously mentioned works use raw similarity scores as meta-features. The use of more elaborated statistics beyond the raw similarity scores was proposed in [3]. In that work, different meta-feature groups were derived from the information provided by class distribution, the entropy and the within-class cohesion observed in the k nearest neighbors of a given test document x.

The combined use of the previously described meta-features [2, 3, 6, 14, 19] has the potential to improve the results of each meta-feature in isolation. However, this combination may be unnecessarily complex and highly dimensional, which increases the tendency of overfitting on classification methods. In order to overcome such problem, [4] proposes an effective but computationaly expensive wrapper feature selection methods designed to select and evaluate groups of meta-features considering the adequacy of the selected meta-features to a particular dataset.

Our proposal differs from [4] on how to combine information from different groups of meta-features, since we aim at avoiding overfitting and costly feature selection by proposing the use of error rate based meta-features capable of informing the classifier about hard-to-classify documents in the meta-level. Moreover, our meta-features aim at measuring the relationships between neighborhood similarity evidence and categories, differently from previous work that only use distance metrics or statistics from them as similarity evidence.

Finally, although not directly related to our proposed approach, as they do not construct enriched representations based on lower-level ones, document representations based on word embeddings such as PTE [17], Fisher Vectors [12] and Paragraph2Vec [11] may serve as an alternative to Bag of Words for creating meta-features, as they exploit potentially useful contextual information. Two drawbacks with this possibility are the fact that embeddings usually rely on external fonts of information that may not be adequate to all tasks and the fact that they are much more expensive to generate than Bag of Words. In any case, we leave this door open for future investigations.

3 DISTANCE-BASED META-FEATURES

Let $\mathcal{X}$ and C denote the input (feature) and output (class) spaces, respectively. Let $\mathbb{D}_{train} = \{(x_i, c_i) \in \mathcal{X} \times C\}|_{i=1}^{n}$ be the training set. The main goal of supervised classification is to learn a mapping function $h : \mathcal{X} \mapsto C$ which is general enough to accurately classify examples $x' \notin \mathbb{D}_{train}$.

3.1 Existing Meta-Features (From the Literature)

Gopal et al. The meta-features proposed in [6] were designed to replace the original input space $\mathcal{X}$ with a new informative and compact input space $\mathcal{M}$. Therefore, each vector of meta-features $m_f \in \mathcal{M}$ is expressed as the concatenation of the following sub-vectors, which are defined for each example $x_f \in \mathcal{X}$ and category $c_j \in C$ for $j = 1, 2, \ldots, |C|$.

- $\vec{v}_{\vec{x}_f}^{neighb} = [dist(\vec{x}_{ij}, \vec{x}_f)]$ A $|C| * k$-dimensional vector whose elements $dist(\vec{x}_{ij}, \vec{x}_f)$ denote a distance score between $\vec{x}_f$ and the i^{th} nearest class c_j neighbor of $\vec{x}_f$ – k is the number of neighbors that belong to the category $c_j \in C$.
- $\vec{v}_{\vec{x}_f}^{cent} = [dist(\vec{x}_j, \vec{x}_f)]$ A $|C|$-dimensional vector where $\vec{x}_j$ is the c_j centroid (i.e., vector average of all training examples of the class c_j).

In [6], the combination of these sub-vectors are generated from three distances (Euclidean, Manhattan and Cosine) to represent documents. The intuition behind these meta-features consists of the assumption that if the distances between an example to the nearest neighbors belonging to the category c (and its corresponding centroid) are small, then the example is likely to belong to c.

Pang et al. Meta-features are centroid distances $\vec{v}^{cent}_{\vec{x}_f}$ generated with cosine similarity. The main goal is to provide compact and informative document representations [14].

Canuto et al. Proposes various groups of meta-features that exploit statistics about the neighborhood [3]. In particular, the first key aspect of these meta-features is the fact that they exploit the continuity hypothesis which guarantees the kNN classifier's success: the existence of a mode in the class distribution of the neighborhood of $\vec{x_f}$ usually determines the category of $\vec{x_f}$. Moreover, they propose a summarized version of the Gopal et al's meta-features [6] through category distance quartiles instead of the full distance distribution, which reduces considerably the number of dimensions, potentially preventing overfitting in small datasets. Another key aspect exploited by these meta-features refers to proximities of the neighbors of $\vec{x_f}$ belonging to some class $c_i \neq c_j$ to the centroid of c_j. This directly evaluates the class cohesion in the neighborhood of $\vec{x_f}$, capturing the uncertainty level in such region of the input space. Finally, the entropy of the neighborhood and the correlation between neighbors from different classes provide additional evidence about the purity of the top ranked neighbors.

Best Comb. The SPEA2SVM strategy, described in [4], is a wrapper to select meta-features effectively. The authors used SPEA2SVM to find which meta-features should be removed from the full set of all previously mentioned meta-features, while maximizing the effectiveness for each dataset. Though computationaly expensive, in some datasets SPEA2SVM presented results comparable to the brute-force strategy, allowing one to evaluate and select only a very small fraction of all possible meta-feature combinations. Thus, "Best Comb." is the most effective combination of meta-features found from all meta-features described in this section. We use it as our strongest baseline.

3.2 Newly Proposed Meta-Features

In here, we provide the necessary details for building our newly proposed meta-features. We first describe how SDRs are built followed by the process of building the first type of meta-features derived from these SDRs. Next, we show how to extend the original set of meta-features proposed in the literature to generate even more discriminative ones. Finally, we detail a set of meta-features designed to evaluate the discriminative information produced by the two previous types of meta-features. Such meta-features are useful to identify hard-to-classify documents, being of great importance when learning robust high quality classification models.

3.2.1 SDRs. Classification effectiveness based on meta-features significantly depends on the similarity evidence used to evaluate different pairs of documents. In order to further advance their potential, we propose to take advantage of the discriminative similarity evidences explicitly captured by SDRs.

Similarity evidences are formally defined as follows. Let $\mathcal{X}$ and $\mathcal{S}$ denote the bag-of-words feature space and the SDR feature space, respectively. The similarity evidence corresponding to the pair of documents $\vec{x}_a, \vec{x}_b \in \mathcal{X}$ is denoted by the SDR $\vec{s}_{ab} \in \mathcal{S}$ expressed as the concatenation of the sub-vectors below:

- $\vec{v}^{common} = [min(\vec{x}_a, \vec{x}_b)]$: A $|\mathcal{X}|$-dimensional vector s.t. each element w in $\vec{v}^{common}$ corresponds to $min(\vec{x}_{a_w}, \vec{x}_{b_w})$, where $\vec{x}_{a_w} \geq 0$ and $\vec{x}_{b_w} \geq 0$ correspond to the TFIDF weights of word w in documents $\vec{x}_a$ and $\vec{x}_b$, respectively[1]. This vector provides a new sparse representation that corresponds to the common information among two documents. This high-dimensional, fine-grained information might identify important individual common similarity evidence that appear in both documents.
- $\vec{v}^{cos} = [\cos(\vec{x}_a, \vec{x}_b)]$: A 1-dimensional vector produced by the cosine similarity between $\vec{x}_a$ and $\vec{x}_b$.
- $\vec{v}^{cent} = [min(\cos(\vec{x}_a, \vec{x}_j), \cos(\vec{x}_b, \vec{x}_j))]$: A $|C|$-dimensional vector formed by the minimum cosine similarities between $\vec{x}_a$ or $\vec{x}_b$ and each one of the x_j category centroids in the training dataset. It captures explicit similarity evidence that relates both $\vec{x}_a$ and $\vec{x}_b$ documents to categories.

Since most documents in a given set of training documents $\mathbb{D}_{train}$ usually do not present meaningful similarity evidence with a target document t, SDRs are built using only the k nearest neighbors of t in $\mathbb{D}_{train}$. Algorithm 1 describes the construction of SDRs for t. It receives t, k and $\mathbb{D}_{train}$ as input and returns a set of SDRs S that represents the similarity evidence found on neighbors of t in a given set of documents $\mathbb{D}_{train}$. In Line 2, the algorithm finds the neighbors of t using the cosine similarity, which presented the best results for meta-feature generation in textual data [4]. Then, for each neighbor, the algorithm uses function $SyntheticDocumentRepresentation(t, n)$ in Line 4 to build a SDR for the pair (t, n) with the previously described similarity features. Therefore, the similarity evidence found on each neighbor is explicitly represented as features from its corresponding SDR.

Algorithm 1: *BuildSyntheticNeighbors*$(t, k, \mathbb{D}_{train})$

Input: Target document t, number of neighbors k and set of documents $\mathbb{D}_{train}$
Output: SDRs S for t

1 $S \leftarrow \emptyset$
2 $N \leftarrow k$ nearest neighbors of $\vec{x}_t$ in $\mathbb{D}_{train}$
3 **foreach** $\vec{n} \in N$ **do**
4 $\quad \vec{s}_{tn} = SyntheticDocumentRepresentation(\vec{x}_t, \vec{n})$
5 $\quad S \leftarrow S \cup \{\vec{s}_{tn}\}$
6 **end**

3.2.2 Meta-Features based on SDRs (SYN). After providing explicit similarity evidence in the form of SDRs, it is possible to learn a predictor that correlates the similarity evidence found in the pair of documents corresponding to a SDR s with the likelihood of these documents belonging to the same class. Thus, the predictor is able to estimate the relevance of similarity evidence to build more informed meta-features for effective text classification.

Algorithm 2 details how the training samples are processed to learn a predictor for SDRs. In Lines 1-2, the algorithm prepares the training data for the generation of a SVM hyperplane hw_c for each category c considering each training

[1] Whenever word w does not occur in a document $\vec{x}_a$, $\vec{x}_{a_w} = 0$.

document $d \in \mathbb{D}_{train}$. Lines 5-7 generate SDRs that are positive training examples related to d using the subset of training examples $\mathbb{D}_{pos}$ of the same category as d. Note that there is no "else" after line 7, since we want to include negative training examples for documents of category c. In other words, these lines produce a set of SDRs $\mathbb{S}_{pos}$ with Algorithm 1 only for pairs of training documents that belong to the same class. On the other hand, Lines 8-9 produce a set of SDRs $\mathbb{S}_{neg}$ only for pairs of training documents that belong to different classes. Finally, in Line 11 a SVM classifier is trained using those positive and negative training samples ultimately defining a separating hyperplane. Such hyperplane is used as a predictor to estimate the likelihood of the similarity evidence in a SDR being related to category c.

Algorithm 2: Global hyperplanes for SDRs.

```
   Input: Training set D_train
   Output: Hyperplanes hw_c for each category c
1  foreach category c do
2      D_pos ← ∅; D_neg ← ∅; S_pos ← ∅; S_neg ← ∅;
3      foreach d ∈ D_train do
4          if category of d = c then
5              D_pos ← documents of category c in D_train
6              S_pos ← S_pos ∪ BuildSyntheticNeighbors(d, k, D_pos)
7          end
8          D_neg ← docs. that are not of category c in D_train
9          S_neg ← S_neg ∪ BuildSyntheticNeighbors(d, k, D_neg)
10     end
11     hw_c ← TrainSVM(S_pos, S_neg)
12 end
```

After learning the hyperplanes hw_c, we use the distances among SDRs and the hypeplanes as meta-features that measure the similarity evidence between a target document and its neighbors. Algorithm 3 describes how meta-features $\vec{m}_t$ are generated for a target document t with the previously built hyperplanes hw_c. Lines 3-4 build the SDRs from the neighbors of t that belong to the training documents of category c. Using the generated SDRs, the method *ComputeHyperplaneDistance* in Line 7 computes the normalized distance (with sigmoid function [15]) between each SDR $s \in S_c$ and the hyperplane hw_c. Such normalized distances correspond to the likelihood of the similarity evidence in each SDR in S_c being related to category c. It is worth noting that each SDR s corresponds to the similarity evidence between the target document t and one of its neighbors. Therefore, a high hyperplane distance between s and hw_c corresponds to a high likelihood of t being related to c.

In Line 10, all the computed distances stored in the set H are sorted in ascending order, generating meta-features in $\vec{m}_{tc}$. This allows a learning method to compare the i-th greatest meta-feature value related to hperplane c of a document to the i-th greatest meta-feature value regarding the same hyperplane of another document during the learning process. Finally, in Line 11, the computed meta-features for class c are concatenated with the output vector $\vec{m}_t$ that contains meta-features for all categories.

It is important to notice that Algorithm 3 is used directly only for the corresponding SDRs of a test example. If applied to SDRs generated for documents in the training set $\mathbb{D}_{train}$, the meta-features generated from these SDRs would be biased to the training data, which consequently overfits the classifier. In order to generate meta-features for training documents, it is necessary to apply Algorithm 3 with cross-validation in the training set, where the hyperplanes hw_c are built from a subset of the training data, and the meta-features are generated for documents in the remaining subset.

Algorithm 3: Building Meta-features from SDRs using Global Hyperplanes (Synglob).

```
   Input: Hyperplanes hw_c, target document t, training set D_train
   Output: Meta-features m_t for the document t
1  m_t ← []
2  foreach category c do
3      D_pos ← documents of category c in D_train
4      S ← BuildSyntheticNeighbors(t, k, D_pos)
5      H ← ∅
6      foreach s ∈ S do
7          h_dist ← ComputeHyperplaneDistance(s, hw_c)
8          H ← H ∪ {h_dist}
9      end
10     m_tc ← sort(H)
11     m_t ← concatenate(m_t, m_tc)
12 end
```

As previously mentioned, we also propose a version of our meta-features inspired on the SVM-kNN method [20], which builds local hyperplanes using only the nearest SDRs of a target document as training examples. In this scenario, the hyperplane construction ignores potentially unrelated documents beyond the neighborhood of a document by locally adjusting the capacity of the SVM hyperplane to the properties of the training set in each area of the input space of SDRs. The construction of meta-features using such hyperplanes is illustrated in Algorithm 4, which differs from the Algorithm 3 by the fact that it builds each hyperplane using only the neighborhood of each target document, as illustrated in Lines 3-7. The construction of one hyperplane for each category of each target document is feasible because of the reduced number of training elements (only the neighbors). This naive implementation can be further improved with the combined use of kernel trick and DAGSVM [20]. Then the algorithm generates one meta-feature for each distance between the target document and the locally built hyperplane in Lines 8-9. The fact that SDRs are generated from training documents of all categories assures the evaluation of similarity evidence found on the relationship between t and neighbors from each category even on unbalanced training datasets.

3.2.3 Extended version of literature Meta-features (EXT). We extend the literature meta-features in two different ways. The first strategy extends the centroid distances previously defined in Section 3.1 with the vector $\vec{v}^{cent}_{\vec{x}_f}$ by evaluating the neighborhood in a projected meta-feature space. The second strategy exploits the space of original features and some literature meta-features using a SVM classifier.

In our extensions, we focus on the two groups of literature meta-features which correspond to the vectors $\vec{v}^{neighb}_{\vec{x}_f}$ and $\vec{v}^{cent}_{\vec{x}_f}$ (Section 3.1) built from the cosine similarity. From now

Algorithm 4: Building Meta-features from SDRs using Local Hyperplanes (Synloc).

Input: Target document t, training set $\mathbb{D}_{train}$
Output: Meta-features $\vec{m}_t$ for the document t
$\vec{m}_t \leftarrow []$
foreach *category* c **do**
  $\mathbb{D}_{pos} \leftarrow$ documents of category c in $\mathbb{D}_{train}$
  $\mathbb{S}_{pos} \leftarrow BuildSyntheticNeighbors(t, k, \mathbb{D}_{pos})$
  $\mathbb{D}_{neg} \leftarrow$ docs that are not of category c in $\mathbb{D}_{train}$
  $\mathbb{S}_{neg} \leftarrow BuildSyntheticNeighbors(t, k, \mathbb{D}_{neg})$
  $hw_c \leftarrow TrainSVM(\mathbb{S}_{pos}, \mathbb{S}_{neg})$
  $h_{dist} \leftarrow ComputeHyperplaneDistance(t, hw_c)$
  $\vec{m}_t \leftarrow concatenate(\vec{m}_t, h_{dist})$
end

on, we call these meta-features, respectively, as *Cos_neigh* and *Cos_cent*. These two meta-features were pointed out in [4] as compact and relatively effective for most datasets.

Centroid-based meta-features (*Cent_ext*). Our extension of centroid meta-features evaluates the neighborhood of documents in a low-dimensional space spanned by class centroids. Due to the already strong discriminative power of class centroids, our extension aims at enhancing them to better handle issues related to class imbalance and noisy terms in the original textual data representation. The strategy to extend centroid meta-features relies on two steps. In the first step, we replace the original input space $\mathcal{X}$ with a new space $\mathcal{M}_{cent}$ corresponding to the *Cos_cent* meta-features. By doing so, documents that were represented as a bag-of-words are then represented as the compact set of centroid distances between the original document and each category centroid.

In the second step, we evaluate the neighborhood of each projected document $\vec{m} \in \mathcal{M}_{cent}$ using the same strategy described in Section 3.1 to generate the distance vectors $\vec{v}^{neighb}_{\vec{x}_f}$. Therefore, we generate meta-features that correspond to the Euclidean distance between a projected document document $\vec{m} \in \mathcal{M}_{cent}$ and each neighbor in the projected meta-feature space of centroids. In other words, we generate a vector $Cent\_ext = [dist(\vec{m}_i, \vec{m})]$, which is a $|C| * k$-dimensional vector whose elements $dist(\vec{m}_i, \vec{m})$ denote the euclidean distance between $\vec{m}$ and the i^{th} nearest class c_j neighbor of $\vec{m}$. The evaluation of the distribution of distances among neighbors in the projected space enables a deeper exploitation of centroids distances, since it takes into account the relationships between close centroids distances from different documents.

Original features and meta-features (*Orig_ext*). Inspired by the success of previous works [3] in combining the high-dimensional original input space $\mathcal{X}$ with literature meta-features, we here propose a compact set of meta-features capable of embodying the main benefits of such combination. This combination, named $\mathcal{X}_{MFextend}$, is an extended feature space that represents documents with the concatenation of their original document representation with meta-features *Cos_neigh* and *Cos_cent*. The extended space $\mathcal{X}_{MFextend}$ enables the classifier that operates in such space to find interesting connections/relationships between meta-features and individual words. In this sense, $\mathcal{X}_{MFextend}$ combines the best of two worlds: summarized discriminative evidence about documents in the form of meta-features and very specific information about the individual words of the documents. In order to build a compact set of meta-features that exploits the relationship between the original and meta-features, we propose to exploit the distances between documents $\vec{x} \in \mathcal{X}_{MFextend}$ and SVM hyperplanes trained to categorize such documents. The resulting hyperplanes discriminate documents $\vec{x}$ according to the information from features and meta-features, which allows us to automatically evaluate the relationships between the two types of features in $\mathcal{X}_{MFextend}$. Particularly, we use a training set to generate one hyperplane per category. The meta-features for a test document are the normalized distances between the document and each hyperplane[2].

In sum, let $hwdist_c(\vec{x})$ be the hyperplane distance between a document $\vec{x} \in \mathcal{X}_{MFextend}$ and the hyperplane trained to categorize documents for category c. We define $Orig\_ext = [hwdist_c(\vec{x})]$ as a $|C|$-dimensional vector that contains the distance between $\vec{x}$ and the hyperplanes generated for each category $c \in C$. In order to avoid overfitting when generating these meta-features for training documents, we use cross-validation in the training set, where the hyperplanes are built from a subset of the training data, and the meta-features are generated for documents in the remaining subset.

3.2.4 Error rate based Meta-features (ERR). The main goal of error rate based meta-features is to estimate whether a target document needs additional information that complements meta-features built from SDRs or from our proposed extension of literature meta-features. Such meta-features are described as follows:

Error rate for SDRs (Err_syn). Let $\mathcal{S}_t$ and $\mathcal{S}_{tcorrect}$ denote the SDRs produced for a target document t using Algorithm 1 and let $\mathcal{S}_{tcorrect}$ be the number of correctly classified SDRs evaluated with the previously trained SVM models in Algorithm 2. We define $Err\_syn = [\frac{|S_{tcorrect}|}{|S_t|}]$ as a 1-dimensional vector produced by the proportion of correctly classified synthetic neighbors generated for t. A high proportion of correctly classified SDRs generated for t indicates that there is reliable similarity evidence provided by SDRs to classify it.

Error rate for extended meta-features (Err_ext). Similarly to the error rate of SDRs, we compute the proportion of correctly classified documents in the previously described extended space $\mathcal{X}_{Orig\_ext}$. In this scenario, we define $Err\_ext = [\frac{|N_{tcorrect}|}{|N_t|}]$ as a 1-dimensional vector produced by the proportion of correctly classified neighbors $N_{tcorrect}$ from all neighbors of a target document $\vec{t} \in \mathcal{X}_{Orig\_ext}$.

Discrepancy on literature-extended meta-features (Discr). We also evaluate discrepancies on scores of $\mathcal{X}_{Orig\_ext}$. The main idea is to evaluate how the hyperplane distance of a target document differs from the hyperplane distances of its neighbors. Accordingly, the fact that a target document is as distant from a hyperplane as its neighbors correlates with the reliability of the evidence in $\mathcal{X}_{Orig\_ext}$ for such target

[2]In order to generate meta-features for training documents, it is necessary to apply cross-validation in the training set, where the hyperplanes are built from a sub-set of the training data, and the meta-features are generated for documents in the remaining subset.

document. Considering the vector of hyperplane distances $\vec{v}^{hwdist}_{\vec{x}}$ defined for a document x in Section 3.2.3, we define the discrepancy meta-features as $\vec{v}^{discrepancy} = [\vec{v}^{hwdist}_{\vec{x}} - \vec{v}^{hwdist}_{\vec{x}_{ij}}]$, which is a k-dimensional vector whose elements denote the difference between the hyperplane distance of $\vec{x}$ to a hyperplane and each hyperplane distance of its i^{th} nearest class c_j neighbor of $\vec{x}$.

4 EXPERIMENTAL RESULTS

4.1 Experimental Setup

4.1.1 Textual Datasets. In order to evaluate the meta-level strategies, we consider five real-world textual datasets for topic classification and 18 publicly available datasets for sentiment analysis. For all datasets, we performed a traditional preprocessing task by: removing stopwords with the SMART list, removing terms with low "document frequency (DF)"[3] and using TFIDF term weighting. We provide the detailed description of our datasets in an online appendix[4].

4.1.2 Evaluation, Algorithms and Procedures. The classification results were evaluated using two standard measures: the micro averaged F_1 ($MicroF_1$) and the macro averaged F_1 ($MacroF_1$) [13, 18]. While the $MicroF_1$ measures the classification effectiveness over all decisions (i.e., the pooled contingency tables of all classes), the $MacroF_1$ measures the classification effectiveness for each individual class and averages them. All experiments were executed using a 5-fold cross-validation procedure. The parameters were set via cross-validation on the training set, and the effectiveness of the algorithms running with distinct types of features were measured in the test partition.

To build the hyperplanes necessary to our proposals and evaluate the effectiveness of different groups of features, we adopted the LIBLINEAR [5] implementation of the SVM classifier. As far as we know, SVM is still the state-of-the-art in text classification for addressing the high dimensional and sparse text data, such as Bag of Words, or in reduced, more compact spaces, such as those based on meta-features. We choose to perform a fair comparison by keeping the linear kernel in all experiments, despite the potential benefits of exploiting more complex kernels with our proposals. We leave such possibility for future work.

The regularization parameter was chosen among eleven values from 2^{-5} to 2^{15} by using 5-fold cross-validation within each training dataset. The neighborhood size k for obtaining the meta-features was chosen among five values from 10 to 50 by also using cross-validation within each training dataset.

To compare the average results on our 5-fold cross-validation experiments, we assess the statistical significance of our results by means of a paired t-test with 95% confidence and Holm correction to account for multiple tests. Results in **bold** are statistically superior to others, and multiple results in boldface are not statistically superior to the best of them.

[3]We removed all terms with DF<6).

[4]https://github.com/UFMG-Database-lab/appendix/sigir19appendix.pdf

We would point out that some of the results obtained in some datasets may differ from the ones reported in other works. Such discrepancies may be due to several factors such as differences in dataset preparation[5], the use of different splits of the datasets (e.g., some datasets have "default splits" such as REUT and 20NG[6]), and the use of lexicons for sentiment analysis[7]. We stress that we ran all alternatives under the same conditions in all datasets, using the best traditional feature weighting scheme, using standardized and well-accepted cross-validation procedures that optimize parameters for each of alternatives, and applying the proper statistical tools for the analysis of the results. Our data and codes are available upon request.

		20NG	4UNI	REUT	ACM	MED
Proposed	$macF_1$	**91.4(0.5)**	**74.4(1.8)**	**41.8(1.9)**	**67.3(1.2)**	**78.9 (0.6)**
	$micF_1$	**91.6(0.5)**	**83.0(0.6)**	**79.7(1.0)**	**77.9(0.3)**	**87.8 (0.4)**
Canuto et al [3]	$macF_1$	88.3(0.6)	66.1(2.6)	32.4(2.6)	64.1(1.1)	72.7(0.5)
	$micF_1$	88.5(0.6)	78.9(1.6)	71.5(0.9)	75.5(0.8)	82.5(0.2)
Gopal et al [6]	$macF_1$	89.5(0.5)	60.6(2.7)	**41.7(2.8)**	62.7(1.4)	74.9(0.2)
	$micF_1$	89.8(0.6)	75.6(0.7)	77.9(1.2)	75.6(0.4)	84.2(0.1)
Pang et al [14]	$macF_1$	77.4(0.6)	56.4(1.8)	37.2(1.6)	52.1(1.6)	46.3(1.0)
	$micF_1$	78.3(0.7)	67.6(1.1)	71.8(0.8)	65.0(0.9)	66.3(1.0)
Best Comb. [4]	$macF_1$	89.7(0.6)	66.5(1.4)	**41.5(3.1)**	64.9(1.4)	75.7(0.6)
	$micF_1$	90.0(0.7)	79.9(1.4)	77.4(1.5)	76.3(0.7)	84.4(0.5)
Bag of Words	$macF_1$	87.8(0.2)	60.4(1.0)	29.5(2.1)	61.6(0.4)	76.0(0.2)
	$micF_1$	87.6(0.2)	70.7(0.8)	65.7(0.7)	72.1(0.5)	85.6(0.5)

Table 1: Average effectiveness on different meta-features.

4.2 Experimental Results

4.2.1 Effectiveness Results. We here present the effectiveness results of classifiers trained with meta-features from different literature works and our proposed approach for the tasks of topic classification and sentiment analysis. Considering the topic classification task, Table 1 shows the obtained values of $MacroF_1$ and $MicroF_1$ for our proposal, the traditional Bag-of-words representation and four sets of meta-level features proposed by Canuto et al (Canuto) [3], Gopal et al (Gopal) [6], Pang et al (Pang) [14] and the recently proposed combination of these three literature meta-feature using genetic algorithms to select the best meta-feature combination for each dataset (Best Comb.) [4]. The *Proposed* meta-feature in Table 1 corresponds to the union of the three types of meta-feature spaces proposed in this work.

As it can be seen, our proposed meta-features consistently achieve the best results in **all** evaluated datasets, a remarkable result. This provides evidence that the combination of meta-features described in Section 3 do produce more discriminative information than other distance-based meta-features in the literature, which rely on distance measures not designed to relate pairwise similarity evidence with categories.

The main difference between our proposal and the remaining methods is the identification of strong clues indicating that one particular neighbor contains important similarity

[5]For instance, some works do exploit complex feature weighting schemes or feature selection mechanisms that do favor some algorithms.

[6]We believe that running experiments only in the default splits is not the best experimental procedure as it does not allow a proper statistical treatment of the results

[7]Best results for sentiment analysis include an external source of labeled lexicons

evidence. Such clues include large distances to a hyperplane (high prediction scores), which are most likely not "false positives" [15]. We further exploit the confidence of predictions about similarity information thru ERR meta-features, which provide evidence about the discriminative information in our proposals. In fact, the proposed meta-features improved the results of previous works by Gopal, Canuto, Pang, and the combination of the best of them (Best Comb.) by 13%, 22%, 28% and 12%, respectively.

We now turn our attention to the results obtained for the 18 sentiment analysis datasets. In general, the task of categorizing sentiment datasets is difficult because of the neutral category, which is commonly confused with other categories because arbitrary documents are usually similar to documents associated with positive or negative sentiments. Considering such datasets, Table 2 presents the results obtained with proposed meta-features, traditional Bag of Words and the best combination of literature meta-features [4].

In this scenario, our proposed meta-features were capable of obtaining, again, the best results in **all** 18 datasets, being the only proposal to achieve these remarkable results. The other two baselines do not come even close–they tie with our approach in 7 and 6 datasets, losing in all others. The most expressive gains against Best Comb. were obtained on ss_rew, ss_digg, ss_bbc and yelp_rev by 10%, 9%, 8.8% and 7.4%, respectively. Such datasets contain reviews or news that are usually bigger (i.e., more than 15 words) than the documents from other datasets, with only a few of them representing discriminative words for sentiment classification. In this scenario, the proposed meta-features can enrich the importance of such words using the labeled information.

dataset	Proposed	Best Comb.[4]	Bag of Words
aisopos_ntua	**73.0(3.5)**	69.5(3.3)	71.1(2.9)
debate	**57.6(2.1)**	**56.9(1.3)**	**57.2(1.1)**
en_dailabor	**73.8(1.9)**	68.9(2.1)	71.9(1.8)
nikolaos_ted	**50.5(1.9)**	**52.6(3.6)**	**50.3(1.9)**
pang_movie	**78.0(0.5)**	77.1(0.6)	76.4(1.0)
sanders	**68.5(1.9)**	65.2(1.5)	65.9(1.0)
ss_bbc	**37.1(4.2)**	34.1(4.2)	27.2(0.6)
ss_digg	**45.5(3.2)**	41.7(2.2)	38.2(3.1)
ss_myspace	**46.0(3.2)**	41.7(3.5)	35.0(2.8)
ss_rev	**46.2(2.6)**	42.0(5.7)	41.4(4.7)
ss_twitter	**56.5(1.3)**	**57.1(1.9)**	**55.5(1.6)**
ss_youtube	**58.1(1.1)**	**56.9(1.8)**	54.3(1.8)
stanford_tw	**85.3(2.7)**	**87.1(2.8)**	**85.7(3.4)**
semeval_tw	**61.9(1.6)**	56.8(1.2)	59.2(1.4)
vader_amz	**49.1(0.7)**	**48.4(1.1)**	**48.0(1.2)**
vader_movie	**52.7(0.4)**	52.2(0.3)	51.9(0.6)
vader_nyt	**42.2(1.5)**	**43.4(1.2)**	**42.6(1.3)**
yelp_rev	**94.3(0.1)**	87.8(1.2)	93.8(0.2)

Table 2: Average Macro-F1 on different meta-feature groups.

4.2.2 Group Evaluation. After evaluating the behavior of the combined use of all the proposed meta-features, we further analyze each component of our proposal. In the following analyses, we focus on the topic datasets as they configure a harder task, being larger (in terms of documents and text length) and involving more classes. Our gains against the strongest baseline (Best Comb.) are also larger in these datasets, making the analyses more interesting. Table 3 shows the effectiveness of each meta-feature group in isolation in terms of micF1[8].

[8]Results with macF1 were qualitatively the same and are omitted for the sake of space.

We first turn our attention to the SYN meta-features, which are based on the supervised evaluation of SDRs. In this group, *Synglob* (Section 3.2.2) always perform significantly better than *Synloc* due to the fact that the latter only uses the limited information provided by the nearest neighbors of training examples, while *Synglob* takes advantage of the whole labeled data. *Synglob*, for instance, was the sole winner on 4UNI, achieving the best results among all groups in this dataset. The task of categorizing academic webpages in 4UNI is difficult as the general pages category is commonly mistaken by others. *Synglob* also appears among the top performers in 4 out of 5 datasets (the exception beinng REUT), a very strong and consistent performance. Particularly, in the case of REUT, it contains several classes with just a few training documents (less than 5 for 26 categories), which prevents the exploitation of SDRs in an effective way.

EXT meta-features also obtained high effectiveness using different strategies to exploit information from similarity evidence. Particularly, *Orig_ext* produced high effectiveness results in general achieving the best results among EXT on 4UNI and MED with the combined exploration of original features and meta-features. Despite its benefits, *Orig_ext* suffers from potential generalization errors because of imbalanced classes and small training, as seen in REUT.. Other EXT meta-features, namely *Cent_ext* and *Cos_cent* obtained significantly lower results in most datasets, as they are designed to complement other meta-features by exploiting only the global information related to class centroids.

Considering the ERR meta-features, we can see that they obtained the lowest results in general. In fact, they were designed to identify hard-to-classify documents based on meta-features, being not explicitly designed to provide direct evidence for classification. As such, their largest benefit should be observed when used in conjunction with other groups.

Since the groups SYN, EXT and ERR were designed to explore different aspects and idiosyncrasies of the text classification task, we expect the presence of complementary information among them. In fact, there is clear empirical evidence to support such complementarity hypothesis. This is best seen with the combination SYN+EXT+ERR, which provides statistically significant superior results when compared to all other possible combinations of SYN, EXT and ERR in all datasets, as shown in Table 4.

Particularly, the comparison between SYN+EXT+ERR and SYN+EXT highlights the importance of ERR meta-features to identify potentially hard-to-classify examples. Such identification provides means for a better optimization process during learning, improving the ability of SYN or EXT for categorizing hard-to-classify examples. This can also help mitigating potential noise from such examples during the model construction.

Disregarding ERR, the combination EXT+SYN is always superior to EXT or SYN in isolation. In fact, SYN and EXT exploit similarity evidence using different methods. Particularly, SYN meta-features take advantage of SDRs to directly

	SYN		EXT				ERR		
Dataset	*Synglob*	*Synloc*	*Cent_ext*	*Orig_ext*	*Cos_neigh*	*Cos_cent*	*Err_syn*	*Err_ext*	*discrepancy*
4UNI	**79.0(1.5)**	64.5(1.0)	71.6(0.9)	75.5(1.0)	71.4(0.6)	70.4(0.4)	45.1(1.2)	45.4(1.1)	69.0(1.3)
20NG	**88.8(0.9)**	72.7(0.7)	78.6(0.6)	87.7(0.2)	**88.5(0.6)**	81.1(0.4)	5.9(0.3)	5.3(0.1)	63.9(1.2)
ACM	**73.9(0.5)**	62.3(0.8)	68.9(0.6)	**74.3(0.4)**	**74.3(0.5)**	70.0(0.3)	24.3(0.5)	26.3(0.3)	60.7(0.4)
REUT	64.7(3.2)	56.8(1.3)	**76.3(0.6)**	69.9(1.5)	**76.1(0.7)**	74.7(0.7)	30.1(0.6)	29.7(0.5)	61.6(0.9)
MED	**84.9(0.6)**	72.9(0.4)	80.3(0.3)	**84.5(0.7)**	82.9(0.9)	79.9(0.3)	51.1(0.5)	52.9(0.4)	75.2(1.0)

Table 3: Average Micro-F1 effectiveness on each group of proposed meta-features.

		20NG	4UNI	REUT	ACM	MED
EXT+SYN+ERR	$macF_1$	**91.4(0.5)**	**74.4(1.8)**	**41.8(1.9)**	**67.3(1.2)**	**78.9 (0.6)**
	$micF_1$	**91.6(0.5)**	**83.0(0.6)**	**79.7(1.0)**	**77.9(0.3)**	**87.8 (0.4)**
SYN+ERR	$macF_1$	89.1(0.5)	70.4(3.3)	34.8 (2.1)	63.5(1.1)	76.5 (0.4)
	$micF_1$	89.3(0.4)	80.0(1.3)	71.8 (2.3)	75.6(0.4)	86.0 (0.3)
EXT+ERR	$macF_1$	88.3(0.4)	65.5(1.0)	36.8 (1.2)	63.1(1.5)	75.9 (0.4)
	$micF_1$	88.5(0.3)	78.7(0.7)	76.9 (0.8)	75.6(0.5)	85.9 (0.2)
EXT+SYN	$macF_1$	90.6(0.4)	70.5(2.0)	**39.4(0.6)**	64.4(0.9)	75.5 (0.7)
	$micF_1$	90.8(0.4)	80.2(0.8)	77.9(0.5)	77.0(0.5)	86.8 (0.5)
SYN	$macF_1$	88.3(0.7)	67.1(3.6)	25.8 (2.5)	59.4(0.6)	75.3 (0.7)
	$micF_1$	88.5(0.7)	79.2(1.5)	65.9 (2.9)	74.5(0.6)	85.1(0.5)
EXT	$macF_1$	88.2(0.2)	65.0(0.9)	37.3 (0.9)	63.7(0.7)	74.3 (0.7)
	$micF_1$	88.3(0.3)	78.6(0.8)	77.0 (0.7)	75.5(0.4)	85.7 (0.3)
ERR	$macF_1$	64.2(0.9)	49.1(2.3)	22.5(1.7)	41.2(1.1)	47.4 (1.3)
	$micF_1$	64.6(0.9)	68.3(1.3)	62.7(1.1)	60.6(0.6)	75.1 (1.2)

Table 4: Average effectiveness on each meta-feature group.

express the relationship between categories and neighborhood-based similarity evidence. On the other hand, EXT meta-features summarize all the similarity evidence with cosine scores without exploiting the other components of the SDRs, especially the relationships with categories. Such significant (and complementary) differences on strategies to exploit similarity information justify the consistent and statistically significant gains ranging from 1% to 5% of EXT+SYN over the best results found either in EXT or SYN.

4.2.3 Importance of Groups with using $2^k r$ Factorial Design . We further analyze the importance of the three groups of meta-features, as well as their interactions, to explain the current results, using all 2^k possible combinations of groups for each dataset. We consider the case without any group using a "random" classifier, which returns an arbitrary category for each document. We also consider the replication of the experiments with each possible combination (using 5-fold cross validation) to evaluate the effects of uncontrollable external factors. We follow the standard quantitative approach called $2^k r$ factorial design [7] to analyze the effects of the individual groups of meta-features, as well as the effectiveness improvements produced by their interactions.

Table 5 presents the percentage of variation in the results that can be explained by each individual group of meta-features and by the interaction between groups of meta-features considering each possible combination. As we can see, the variations observed on all combinations can mostly be explained by the groups SYN and EXT in isolation and the iteration SYN:EXT. The effects of SYN and EXT each always account for more than 23% of all the $MicroF_1$ variation[9], as the presence of each one of these groups in isolation provides discriminative information for the text classification task. The iteration SYN:EXT also explains up to 21% of all variations in the results, highlighting the complementarity between SYN and EXT. Altogether, SYN, EXT and SYN:EXT explain more than 70% of the results in all analyzed datasets.

[9]Again, macF1 results are equivalent.

Other measured effects that present the interaction of ERR with other groups consistently explain statistically significant portions of the variation in the results, ranging from 4% to 8%. Such consistent variations, though relatively small in isolation, account altogether for about 17% of the total variation in the results. This provides strong evidence of the importance of ERR to improve the results of other meta-feature groups when interacting with them. But even in isolation, ERR can explain a rather interesting portion of the results (between 7%-8%). Finally, the residuals (the inexplicable fraction of the variation) are quite low, meaning that we can safely ignore external factors beyond EXT, SYN and ERR.

	20NG	4UNI	REUT	ACM	MED
SYN	25.63	29.03	11.34	25.28	23.58
EXT	24.89	26.13	48.49	27.29	36.89
ERR	7.34	7.68	7.49	7.53	8.16
SYN:EXT	21.15	18.77	9.73	20.29	16.94
SYN:ERR	6.65	5.06	7.14	6.04	3.52
EXT:ERR	7.37	6.46	7.22	7.15	6.79
SYN:EXT:ERR	6.94	5.94	7.59	6.33	4.12
Residuals	0.03	0.94	0.98	0.09	0.01

Table 5: Explained percentage of result variation by individual meta-feature groups and interactions between them. The 95% confidence intervals are always inferior to 0.5%.

5 CONCLUSIONS

Inspired by the success of distance-based meta-feature representations in text classification, in this paper we propose a novel set of new meta-features that tackle some limitations or extend existing proposals in the literature. Our new meta-features based on SDRs aim at explicitly capturing the relationships between document distances and labeled information. The Extended meta-features explore potential relationships that do exist between the literature meta-features and the original words used to build them. Finally, the ERR meta-features aim at identifying hard-to-classify documents, complementing other meta-feature representations and allowing classifiers to adjust the learning process, as this information is now part of the document representation itself. This ultimately improves generalization and mitigates undesired effects of noise. Through a detailed and carefully designed set of experiments, we show that our proposal achieves significant gains of more than 12% against the best combination of meta-features found in the literature in all considered datasets. Our group and factorial analyses show that the meta-features groups we propose do indeed provide complementary information to each other, producing their best results when used altogether.

ACKNOWLEDGMENTS

Partially supported by CNPq, Capes and Fapemig.

REFERENCES

[1] Sergio Canuto, Marcos André Gonçalves, and Fabrício Benevenuto. 2016. Exploiting New Sentiment-Based Meta-level Features for Effective Sentiment Analysis. In *WSDM*. ACM, 53–62.

[2] Sergio Canuto, Goncalves Marcos, Wisllay Santos, Thierson Rosa, and Martins Wellington. 2015. Efficient and Scalable MetaFeature-based Document Classification using Massively Parallel Computing. In *SIGIR*. 333–342.

[3] Sergio Canuto, Thiago Salles, Marcos André Gonçalves, Leonardo Rocha, Gabriel Ramos, Luiz Gonçalves, Thierson Rosa, and Wellington Martins. 2014. On Efficient Meta-Level Features for Effective Text Classification. In *CIKM*. 1709–1718.

[4] Sergio Canuto, Daniel Xavier Sousa, Marcos Andre Goncalves, and Thierson Couto Rosa. 2018. A Thorough Evaluation of Distance-Based Meta-Features for Automated Text Classification. *IEEE Transactions on Knowledge and Data Engineering* (2018), 1–1. https://doi.org/10.1109/tkde.2018.2820051

[5] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *JMLR* 9 (2008), 1871–1874.

[6] Siddharth Gopal and Yiming Yang. 2010. Multilabel classification with meta-level features. In *Proc. SIGIR*. 315–322.

[7] Raj Jain. 1991. *The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling.* Wiley. I–XXVII, 1–685 pages.

[8] Antonia Kyriakopoulou and Theodore Kalamboukis. 2007. Using clustering to enhance text classification. In *SIGIR'07*. 805–806.

[9] Antonia Kyriakopoulou and Theodore Kalamboukis. 2007. Using Clustering to Enhance Text Classification. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '07)*. ACM, New York, NY, USA, 805–806. https://doi.org/10.1145/1277741.1277918

[10] A. Kyriakopoulou and T. Kalamboukis. 2008. Combining Clustering with Classification for Spam Detection in Social Bookmarking Systems *(RSDC '08)*.

[11] Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32 (ICML'14)*. JMLR.org, II–1188–II–1196. http://dl.acm.org/citation.cfm?id=3044805.3045025

[12] Guy Lev, Benjamin Klein, and Lior Wolf. 2015. In Defense of Word Embedding for Generic Text Representation.. In *NLDB (Lecture Notes in Computer Science)*, Chris Biemann, Siegfried Handschuh, AndrÃ© Freitas, Farid Meziane, and Elisabeth MÃ©tais (Eds.), Vol. 9103. Springer, 35–50. http://dblp.uni-trier.de/db/conf/nldb/nldb2015.html#LevKW15

[13] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *JMLR*. 5 (2004), 361–397.

[14] Guansong Pang, Huidong Jin, and Shengyi Jiang. 2015. CenKNN: a scalable and effective text classifier. *DMKD* 29, 3 (2015), 593–625.

[15] John C. Platt. 1999. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*. MIT Press, 61–74.

[16] Bhavani Raskutti, Herman L. Ferrá, and Adam Kowalczyk. 2002. Using Unlabelled Data for Text Classification through Addition of Cluster Parameters. In *ICML'02*. 514–521.

[17] Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. PTE: Predictive Text Embedding Through Large-scale Heterogeneous Text Networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15)*. ACM, New York, NY, USA, 1165–1174. https://doi.org/10.1145/2783258.2783307

[18] Yiming Yang. 1999. An Evaluation of Statistical Approaches to Text Categorization. *Inf. Ret.* 1 (1999), 69–90. Issue 1-2.

[19] Yiming Yang and Siddharth Gopal. 2012. Multilabel classification with meta-level features in a learning-to-rank framework. *JMLR* 88 (2012), 47–68. Issue 1-2.

[20] Hao Zhang, Alexander C. Berg, Michael Maire, and Jitendra Malik. 2006. SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR '06)*. IEEE Computer Society, Washington, DC, USA, 2126–2136. https://doi.org/10.1109/CVPR.2006.301