Classification of Short Texts by Deploying Topical Annotations*

Daniele Vitale, Paolo Ferragina, and Ugo Scaiella

Dipartimento di Informatica
University of Pisa, Italy
{d.vitale,ferragina,scaiella}@di.unipi.it

Abstract. We propose a novel approach to the classification of short texts based on two factors: the use of Wikipedia-based annotators that have been recently introduced to detect the main topics present in an input text, represented via Wikipedia pages, and the design of a novel classification algorithm that measures the similarity between the input text and each output category by deploying only their annotated topics and the Wikipedia link-structure. Our approach waives the common practice of expanding the feature-space with new dimensions derived either from explicit or from latent semantic analysis. As a consequence it is simple and maintains a compact intelligible representation of the output categories. Our experiments show that it is efficient in construction and query time, accurate as state-of-the-art classifiers (see e.g. Phan *et al.* WWW '08), and robust with respect to concept drifts and input sources.

1 Introduction

Text Categorization (TC) is the problem of labeling natural language texts with one or more thematic categories drawn from a predefined set. It is one of the most important research fields in Information Retrieval (IR), and its solutions are at the core of several applications ranging from the automatic cataloging of newspaper pages and web pages, the management of incoming emails, as well as to the annotation of genomic sequences [15]. The majority of existing text classifiers represent a text as a bag of words (shortly, BOW), and then use machine learning techniques over vectors in a high-dimensional space whose features are reals derived from the occurrence frequencies of those words. Many learning methods, such as k-nearest neighbors (k-NN), Naive Bayes, maximum entropy, and support vector machines (SVM), have been used over BOW to solve a lot of classification problems achieving satisfactory results (see e.g.[15]).

In the last decade, however, the explosion of applications in the field of e-commerce, social networks, search engines, instant messaging and information publishing, introduced a new challenging scenario in which texts are very short (a few tens of terms), sparse and poorly written. These texts do not provide

* This work has been supported in part by MIUR PRIN MadWeb, MIUR FIRB Linguistica '06, and a Google Faculty Award 2010.

R. Baeza-Yates et al. (Eds.): ECIR 2012, LNCS 7224, pp. 376–387, 2012.

enough word co-occurrence or shared context to obtain accurate results via classic similarity text measures, as the ones hinging onto the BOW method, so their performance becomes quite limited in this new scenario.

Recently, several studies tried to overcome these limitations inspired by the observation that when humans approach the task of text categorization, they interpret the specific wording of the document in the much larger context of their background knowledge and experience. This has been technically implemented by using external knowledge bases to endow the algorithms with the breadth of knowledge available to humans. Examples are DMOZ [5], Wikipedia (see e.g. [2,6,12]) or even the whole Web [13]. Two main approaches have been followed to deploy this encyclopedic knowledge: one computes similarity scores between texts based on the results returned by a search engine [2,13]; the other one enriches the BOW representation with new dimensions representing *topics* detected in the input texts. In both cases the classification performance is significantly better than the one obtainable with previously known approaches, and today LSA-based approaches (e.g. [12]) are the state-of-the-art.

Our work belongs to this second line of research, in that it deploys an external-knowledge base (namely Wikipedia) to derive the *topics* which occur in the texts to be classified, but it diverts from the known approaches in many respects that we are going to comment below. First of all, we detect the topics occurring in the input texts by using a recent set of IR-tools, called *topic annotators* [4,8,9,10]. These tools are efficient and accurate in identifying meaningful sequences of terms in a input text and in linking them to pertinent Wikipedia pages representing their underlying topics. Among all known annotators we decided to use TAGME [4] because it is the state-of-the-art for processing short texts. There are several positive issues deriving from this choice, if compared to the state-of-the-art classifiers based on LSA-methods [12]. First, TAGME does an explicit semantic analysis of the input texts by manipulating in a principled and controlled way *manifest topics* grounded in Wikipedia, rather than the *latent topics* derived by using LSA. Second, TAGME is efficient in distilling the used knowledge from the *entire* Wikipedia, unlike LSA-approaches that need to restrict the analysis to a limited set of training data, because of their high time complexity. Finally, TAGME hinges on very few parameters and this helps to improve its generalization performance, unlike the LSA-approaches that use many parameters that need an extensive tuning and an “appropriate universal” training set [12]. The main specialty of our proposal is that we do not use the detected topics (Wikipedia pages) to expand the BOW-space or to enrich the content of the input texts, as done before. Rather, we characterize the output categories via a set of *top topics*, derived from the annotation of TAGME over the training samples. The selection is based on a ranking function that mimics the classical *tf-idf* scheme here adapted to work on topics *vs* categories, rather than terms *vs* documents. The key advantage of this topical-representation of output categories, compared to LSA and derivatives, is that the *top topics* are grounded in Wikipedia pages and thus can be interpreted easily.

Our final contribution is the design of a single-label classifier that is based on a novel similarity measure between the set of *top topics* of a category and the set of topics individuated by TAGME in the input text. This similarity measure takes into account the link-structure of Wikipedia by extending the measure proposed in [11] for a pair of topics to a pair of *sets* of topics. This way our classifier deploys not only the content of Wikipedia pages but also their interlinked structure, which has been proved in the last years to be much useful to detect semantic similarity among short texts (see e.g. [4,11]).

We have tested our approach over 3 datasets, one composed of search-result snippets and made available by [12], the other two datasets are composed of tweets and news and were created by ourselves. On the first dataset (snippets) our classifier yields a performance comparable to the one achieved by the best approach based on pLSA [12]; but, unlike pLSA, our classifier does not need a time-consuming training phase. On the other two datasets (news and tweets) our classifier improves significantly known approaches, based on BOW and machine learning tools, when few training examples are available and it shows to be robust with respect to the concept-drift problem.

2 Related Work

Several studies in the last years tried to handle the problem of classifying short texts. They can be grouped in two main approaches: the first one proposed similarity distances specifically tailored to work on short texts, the second one focused on enriching the BOW representation by generating new features derived from external-knowledge bases, such as Wikipedia.

Sahami and Heilman [13] proposed a similarity kernel function between short texts based on the results returned for a web-search composed according to the two texts to be compared. This function can then be used in any kernel-based machine learning algorithm, such as k-NN. This method (and other similar ones, e.g. [2]) is mainly term-based and time consuming, because it needs to query repeatedly a search engine, and thus it does not fit well with applications managing a stream of short texts such as the ones mentioned in the introduction.

Zelikovitz and Hirsh[19] suggested to find similarity between two documents that can be related but don't share words, one from the training set and the other from the test set, by leveraging their similarity to other unlabeled but longer documents. In a later work, Zelikovitz et al. [20] used techniques of Transductive Latent Semantic Analysis, a variant of LSI, to expand the training set with unlabeled test examples. This idea has been further developed and improved in 2008 by Phan et al. [12]. They presented a framework for discovering hidden topics based on LDA, a variant of pLSA, and then added these topics to the BOW-representation of short texts, which were then classified using a maximum entropy learning method. This approach offers high accuracy over two specific datasets consisting of few MBs of training data (one composed of about 12k web-search results and the other one consisting of 50k medical abstracts). Nonetheless it presents two main limitations: it needs the tuning of many parameters and a high time complexity, so it takes days of computing time even if it is restricted to work on a reduced "universal dataset" that must be properly built (see [12]).

Gabrilovich and Markovitch [6] (see also [1,18]) proposed to perform a semantic analysis based on *manifest topics* grounded in Wikipedia pages, rather than *latent topics* used by LSA. So they expanded the BOW-representation of short texts with new dimensions representing *topics* detected in the input texts to be classified and modeled by Wikipedia pages. The mapping between each text and the Wikipedia topics is achieved through a feature generator which acts like a retrieval engine. It receives a short text and outputs the most relevant Wikipedia pages which are related to that text. The titles of these pages are further filtered and those with high discriminative capacity are used as additional features to expand the BOW-representation of the corresponding input texts. The expanded feature vectors are then classified using SVM. [6] showed significant improvements with respect to learning methods based on the *plain* BOW. Our proposal is also based on Wikipedia pages as topics, but we detect these topics via modern annotators, such as TAGME [4], and we use these topics to provide a novel text/category representation rather than deploying them to expand the BOW-space. These annotators are efficacious in disambiguating polysemous terms and in relating synonymous terms in short texts, so we argue that they should be more powerful than just searching Wikipedia for related pages, as done in [6]. Moreover, the experimental results in [11] show that expanding the BOW-space with possibly noisy topics is much more time consuming, induces a complicated feature post-selection step, and it is less effective than just *directly comparing* Wikipedia pages by means of the inter-linked structure of the Wikipedia graph, as we do in our paper.

Recently Sriram et al. [16] proposed an algorithm specifically designed to classify Twitter messages in a set of five categories: news, events, opinions, deals and private messages. The classification task was based on eight features which were domain specific. A more general framework for Twitter messages is described in [7]. The authors proposed to map tweets to Wikipedia pages and then to calculate a distance between them by measuring the overlap of their Wikipedia categories. But it is well known that the category graph of Wikipedia is "haphazard, redundant, incomplete, and inconsistent" [10]; moreover the searches of [7] between single words and titles of Wikipedia pages may possibly miss a *multi-term* topic.

The softwares used in the above papers are missing, so we will use only the results reported in [12] for which the dataset is available. Another contribution of our paper is to make publicly available two large datasets (tweets and news) and the software of our classifier for future (repeatable) comparisons.

3 Topic Annotators

A recent line of research [4,8,9,10] has started to successfully address the problem of providing a semantic contextualization of texts by detecting short and meaningful sequences of terms into them and then link each sequence to a relevant and pertinent Wikipedia page (aka topic). These links solve efficaciously synonymy and polysemy issues, because the identified Wikipedia pages unambiguously represent the specific topics denoted by those sequences of terms given the context offered by the input text. As an example, let us consider the following text fragment: “US president issues Libya ultimatum”. These topic annotators are able to detect “US president”, “Libya” and “ultimatum” as meaningful phrases to be linked with the topics represented by the Wikipedia pages dealing with the President of the United States, the nation of Libya and the threat to declare war, respectively. This contextualization is very powerful because it may help in detecting the semantic similarity of texts not sharing terms, which is one of the limitations of the classical similarity measures based on the BOW-models. Indeed, consider this text fragment: “Barack Obama says Gaddafi may wait out military assault". It would be difficult to detect the tight relationship between this one and the previous text by using classical similarity measures based on word matches, *tf-idf* or co-occurrences. On the contrary, the concepts associated to the input texts by topic annotators might allow one to discover easily this connection by taking into account the Wikipedia link-structure (more later).

The disambiguation task performed by these annotators also prevents errors due to ambiguous words. For example consider the following two similar texts: “the paparazzi photographed the star" and “the astronomer photographed the star". A word-based approach would find hard to figure out their wide topic-distance. Topic annotators instead would link the word “star" in the first fragment to the Wikipedia page entitled “Celebrity" and, in the second fragment, to the page “Star"(intended as the astronomical object). And since these two pages (topics) are far in the Wikipedia graph, an algorithm could easily spot the semantic distance between the two phrases.

4 Our Topic-Based Classifier

Unlike previous works that aimed for expanding the BOW-representation with features extracted from external knowledge bases (see Section 2), consisting of either explicit [6] or latent topics [12], our classifier relies only on the topics identified by Tagme in the input texts and on the connectivity among the corresponding Wikipedia pages in the underlying Wikipedia graph. As an example, recall the previous two phrases about US President and Obama. Previous classifiers would relate these phrases by expanding “Barack Obama” and “US President” with all possible related topics, hoping to find the common topic “President of the United States” among them. Conversely, we process the two phrases with Tagme and thus annotate the segments “Barack Obama” and “US President” with their corresponding Wikipedia pages. At this point we can detect on-the-fly the strong relationship between those two texts by identifying the proximity in the Wikipedia graph of these two Wikipedia pages.

Starting from these promising considerations, we have designed our short-text classifier to work as follows. We annotate all training samples with Tagme and then characterize each output category via a set of *top topics* chosen by a combination of their frequency in the training samples and their diversification.

At query time, the input text is annotated by TAGME and its set of topics is compared against the set of top-topics of each category, searching for the most similar one. The similarity is computed by extending in a principled way the measure devised in [11] for just a pair of topics to a pair of *sets of topics* taking advantage of the Wikipedia link-structure. Clearly our classifier does not rely on any learning method— such as k-NN, Naive Bayes, or SVM. An extensive set of experiments will show in Section 5 that this classification framework is efficient, accurate and robust in that it can be applied successfully to news, tweets and snippets, and because it can address the concept-drift problem [14] without the need to change significantly its parameters and structure.

4.1 The Training Phase

Let $\mathcal{C}$ be the set of known categories, we use TAGME to process all training samples and, for each category $c \in \mathcal{C}$, we denote by $\mathcal{T}_c$ the whole set of topics identified in the samples labeled as c. Then we apply a properly defined (see below) ranking function to each topic in $\mathcal{T}_c$, and finally select the top-k topics that will be used to characterize the category c in the subsequent testing phase. In the rest of this paper we will denote by $\mathcal{T}_c^k$ the set of top-k topics selected for the category c. As the experiments will show in the next Section 5, the only parameter we have to control in this process is k.

The key issue for implementing the training phase is therefore the design of the ranking function which helps in selecting the top-k topics in $\mathcal{T}_c$. This function is inspired by the well-known *tf-idf* schema, here transposed to work in the context of topics vs categories, rather than terms vs documents. For each category $c \in \mathcal{C}$, we define: $rank_c(t) = freq(t, c) \times \log \frac{|\mathcal{C}|}{C(t)}$, where $freq(t, c)$ is the number of training documents belonging to category c and annotated with topic t by TAGME, whereas $C(t)$ is the number of categories whose samples have been annotated with t (hence $t \in \mathcal{T}_c$). We can paraphrase this formula by saying that the first part depends on the topic frequency, whereas the second part depends on the inverse category frequency of t. The idea is to penalize topics that are very common among categories and to rank higher topics that are the most discriminative for the category c.

The best k topics in $\mathcal{T}_c$ are selected in accordance to $rank_c$ and they define the classification model for the category c. Parameter k has to be chosen carefully because it affects the time efficiency of the classification process and its accuracy. A large k could include too generic topics, while a small k could reduce the generalization performance of the model. The value of this parameter will be analysed in the experimental section.

4.2 The Classification Phase

Given an input short text d, we use TAGME to detect the set of topics mentioned in it, say $\mathcal{T}_d$. We then compute a classification score for each category $c \in \mathcal{C}$ by *comparing in a proper way* the topics discovered by TAGME in d, hence $\mathcal{T}_d$, and the top-k topics modeling category c, hence $\mathcal{T}_c^k$. The key issue here is how to "compare" these two sets of topics in order to establish how much *related* are the short text d and the category c. Looking carefully at the problem we are indeed required to compare two sets of nodes (pages of Wikipedia) living into a much larger graph (Wikipedia graph), so we could adopt any *distance* function based on the structure and inter-linkedness of that graph.

There have been many proposals of distances between Wikipedia nodes (see e.g [1,6,11,17]). The most effective one is currently the Wikipedia Link-based Measure (WLM) proposed by Milne and Witten in [11] and extensively tested in [10]. Technically, WLM computes the relatedness between two topics as a function of the size of the intersection between their ingoing stars in the Wikipedia graph. In other words, the relatedness is estimated by taking into account the number of simple 2-long paths connecting the two compared nodes (for details see [11]). This measure finds its theoretical roots in the Google Similarity Distance of [3]. WLM is both cheaper, more robust and more accurate than other known measures: cheaper because Wikipedia's extensive textual content may be ignored in its calculation (cfr. [1,6]), more robust because it computes the *volume* of short paths connecting the compared nodes rather than their shortest-path distance (cfr. [17]), and more accurate because it is more closely tied to the manually defined semantics of the resource.

In our context the comparison between a category c with an input text d boils down to the comparison of two sets of topics, namely $\mathcal{T}_c^k$ and $\mathcal{T}_d$. We therefore extend the WLM measure by computing the *sum of the relatedness* between each pair of topics, one annotated in d and the other one from the top-k topics of category c. Each term of this sum is *weighted* to take into account the fact that topics in d do not have the same importance in characterizing the "category" of this text. We have chosen as weight the score assigned by Tagme to each annotation in d, called ρ-score in [4]. This value measures the importance and reliability of an annotation with respect to all other annotated topics in the input text d, and thus reasonably quantifies how much an annotation "can say" about the category of that input text.

Formally, the classification score of the input text d into the category c has been defined as follows:

$$CSV_c(d) = \sum_{t' \in \mathcal{T}_d} \rho(d, t') \times \left(\sum_{t'' \in \mathcal{T}_c^k} rel(t', t'') \right)$$

where $\rho(d, t')$ is the ρ-score assigned by Tagme to the annotation t' detected in d and $rel(t', t'')$ is the WLM-measure of [11] applied to the two Wikipedia pages denoting the topics t' and t''. The category with the highest score is the classification of d. Note that our classifier differs significantly from known approaches commented in Section 2, because (1) it deploys for the first time the recently proposed topic annotators (here we used Tagme) to detect topics within the input texts to be classified, and, more importantly, because (2) it uses these detected topics not to augment the BOW-representation or the text content of the training and test samples, but rather to directly compare the input text and the output categories via the novel similarity function $CSV_c(d)$ which strongly deploys the connectivity of the Wikipedia graph and no learning methods.

5 Experimental Evaluation

We evaluated our classifier over three datasets composed of snippets of web pages returned by a search engine, short news extracted from RSS feeds, and micro-blogging messages taken from Twitter. The first dataset, called Snippets, was released by [12] and it is composed of 12k (10k of training and 2k of test) snippets returned by a web-search engine. Snippets have length of about 13 terms (on average) and are labeled with 8 categories. This dataset was built to limit the co-occurrence of terms between training and testing samples, and thus to emphasize the performance of semantic-based classifiers as the ones experimented in [12]. It must be said that [12] removed *stop* and rare words from the snippets: this is clearly dis-advantageous for our approach because, unlike the approaches based on pLSA or ESA, Tagme operates at multi-words level and thus could miss some annotations. Nonetheless, to our knowledge, this is the only dataset of short texts available to the community[1] and thus we used it to compare our classifier with respect to the state-of-the-art proposal of [12].

The second dataset, called News, is composed of 32k short texts drawn from the RSS feeds of three newspapers: `nytimes`, `reuters` and `usatoday`. We built it by gathering all news stories published by these editors from March 2011 until June 2011, and took their title and the short abstract, if available, accounting for about 20 terms on average. We derived the category of each news from the taxonomy of its publishing web-site. However, since the three taxonomies are different, we have identified seven common categories: World, US, Science and Technology, Sport, Business, Health, Entertainment.

The last dataset, called Tweets, is composed of about 7k messages that we gathered from Twitter in July 2011 according to the following process: (a) we downloaded from the public stream of Twitter only the messages that contained more than 10 alphabetic chars and a link to some popular web-site of news[2]; (b) we parsed the linked news-page in order to find the category in which that page is classified by the web-site itself; (c) we labeled the Twitter message with this category, limiting to the seven common categories listed above. Here we postulate that the classification label assigned to each tweet is correct because the tweet probably contains a comment or a description of the linked news-story and thus can be categorized in the same category of that linked news. We argue that this dataset is hard to classify because of the well-known poor textual composition of tweets, which amounts to an average of 8 words in our dataset.

We created the training and test sets by splitting in two halves both News and Twitter datasets according to the publishing date. We performed several

[1] We also considered the datasets by [19,20] but those texts are composed of an unordered sequence of stemmed terms. Thus they cannot be processed by the topic annotators presented in Sect. 3 because they need the full-words.

[2] We consider links to `cnn.com`, `huffingtonpost.com`, `nytimes.com` and `reuters.com`

kinds of experiments over these three datasets. The first experiment aimed at evaluating the robustness of the classifier by varying the size of the training set: we trained 10 different classifiers on an increasing number of training samples ranging from 10% to 100% of the overall training data. For each size, we computed the accuracy of the classification process with respect to the test set (of about 16k for News and 3.5k for Tweets). We call this test VarTrain. The second experiment was aimed at evaluating the robustness of classifiers with respect to the concept-drift problem [14], because the meaning of some concepts could change over the time thus defeating the classification model built in advance (this test is called ConceptDrift). This issue is important when dealing with time-related sources like the ones generating our datasets News and Tweets. Thus we divided these two datasets in timely-based partitions, so we used the first week of the training set of News (1054 news) to train the model, and then we generated 7 tests (of about 2300 news on average) by dividing the test set of news in a weekly-basis. Similarly for Tweets, we used the first two days to train the model (738 news), then we discarded the next two days, and finally we generated 6 tests (of about 890 tweets on average) by dividing the remaining tweets on a two-days basis. As can be seen in both cases we introduced a time-gap between the training set and the first test set, in order to limit overlapping topics. For all experiments we reported the classification accuracy. All three datasets are available at `http://acube.di.unipi.it/datasets`.

Parameter Tuning. Our approach relies on just one parameter: the number k of top-topics selected to model each category (see Section 4). For each experiment, we identify the best value of k in the range 10–50 by using a small portion of the training set (about 10%) as validation set. Our experiments showed that, for all datasets and all tests, the best k is small and ranges from 15 to 30 (for higher values the accuracy remains constant or gets slightly worse).

Results for Snippets. This dataset was built by the authors of [12], so its train/test parts are predefined. As pointed out above the available data miss part of the original texts so it is unfavorable for our approach. Nevertheless Fig.1 shows that our classifier comes very close to the accuracy reported in [12][3]. More importantly our classifier can be trained in just 30mins on a commodity PC, whereas [12]'s approach requires several *hours of computing time over a cluster of PCs.* Finally our classifier is faster to be queried because of the compactedness of text and category representations, and the peculiarity of our similarity distance which depends only on the structure of the Wikipedia graph.

Results for News. Since [12]'s system is unavailable, we compared our approach with classifiers based on machine-learning algorithms over the BOW representation: typically they use Bayesian approach (Naive Bayes), tree classifier (C4.5), Support Vector Machine (SVM) or Maximum Entropy (MaxEnt). The features for the BOW-approaches were weighted following the *tf-idf* schema.

[3] Performances of this system and MaxEnt (the baseline in [12]) are extracted from [12] because the software is unavailable. Hence we could not check accuracy of this classifier on our two other datasets.

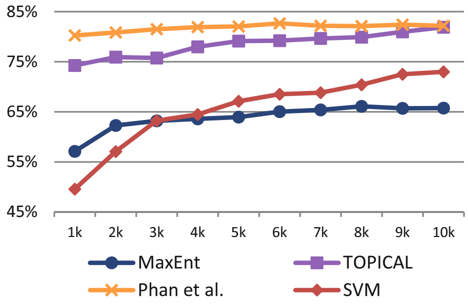
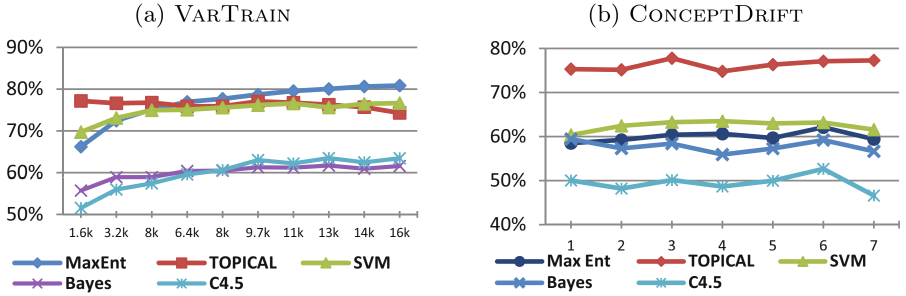


Fig. 1. Evaluation over SNIPPET dataset. TOPICAL is our proposal. On the y-axis is reported the accuracy, on the x-axis is reported the size of training set.

The VARTRAIN test, reported in Fig 2.a, shows an impressive accuracy when classifying new data with a very small number of training samples. Our approach resulted also very accurate for the concept-drift problem, as shown in Fig 2.b. In conclusion, our approach is better when the training set is small with respect to the test set. This can be explained by the fact that this dataset is maybe easy for machine-learning tools since news were collected in a short time window, thus their set of features may be homogeneous. This is different from SNIPPETS, which was created to limit the co-occurrence of terms between the training and the test documents.

Results for TWEETS. This dataset is challenging because the tweets are very short, noisy, and not always in English. However, as shown in Fig. 3, our approach yields an impressive improvement of classification accuracy with respect to the classic BOW-based approaches for both tests. Notice that the overall performance is lower than the one achieved by all classifiers over the other datasets, underlying the fact that this dataset is very difficult because of tweets features. Also in this case, our classifier is better for smaller training sets and this feature


(a) VARTRAIN

(b) CONCEPTDRIFT

Fig. 2. Evaluation over NEWS dataset. TOPICAL is our proposal. On the y-axis is reported the accuracy. On the x-axis of (a) is reported the size of training set, while on the x-axis of (b) are reported the 7 test sets chronologically ordered.

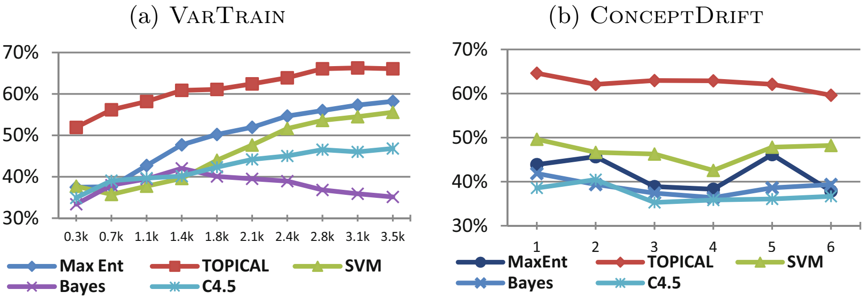


Fig. 3. Evaluation over Twitter dataset. Topical is our proposal. On the y-axis is reported the accuracy. On the x-axis of (a) is reported the size of the training set, while on the x-axis of (b)are reported the 6 test sets chronologically ordered.

is particularly useful in the context of Twitter where is reasonable to argue that the topics of discussion change frequently, so it's more difficult to individuate a representative training set.

6 Conclusions

We presented a novel approach to the classification of short texts which resulted very accurate, efficient, simple and providing a compact representation of categories/texts. Known semantic-based classifiers (see e.g. [6,12]) are instead more complex and time-consuming for the training and the classification phases. Our experiments showed also that our approach is robust with respect to the input source and the concept-drift problem. There are some other issues that we plan to investigate in the near future.

The accuracy of our algorithm depends on the quality of the annotation produced by Tagme [4] and on the relatedness measure introduced by [11]: even though they are shown to be effective, it is worth considering a deeper tuning phase of these tools, or designing variations given the specific TC-task in hand.

Finally, we note that the literature is missing of repeatable experiments because the software implementing the recent solutions of [6,12] is not available. We foresee to re-implement and make available these two approaches, evaluate them over our publicly-available datasets, and thus provide a standard benchmark for future proposals.

References

1. Banerjee, S., Ramanathan, K., Gupta, A.: Clustering Short Texts using Wikipedia. In: ACM SIGIR, pp. 787–788 (2007)
2. Bollegala, D., Matsuo, Y., Ishizuka, M.: Measuring semantic similarity between words using Web Search engines. In: WWW, pp. 757–766 (2007)

3. Cilibrasi, R., Vitanyi, P.: The Google similarity distances. IEEE Trans. on Knowl. and Data Eng. 19(3), 370–383 (2007)
4. Ferragina, P., Scaiella, U.: TAGME: On-the-fly annotation of short text fragments (by Wikipedia entities). In: ACM CIKM, pp. 1625–1628 (2010)
5. Gabrilovich, E., Markovitch, S.: Feature generation for text categorization using world knowledge. In: Int. Joint Conference on A.I, pp. 1048–1053 (2005)
6. Gabrilovich, E., Markovitch, S.: Wikipedia-based Semantic Interpretation for Natural Language Processing. J. Artif. Intell. Res. 34, 443–498 (2009)
7. Genc, Y., Sakamoto, Y., Nickerson, J.V.: Discovering Context: Classifying Tweets through a Semantic Transform Based on Wikipedia. In: Schmorrow, D.D., Fidopiastis, C.M. (eds.) FAC 2011. LNCS, vol. 6780, pp. 484–492. Springer, Heidelberg (2011)
8. Hoffart, J., Yosef, M.A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: Robust Disambiguation of Named Entities in Text. In: EMNLP, pp. 782–792 (2011)
9. Kulkarni, S., Singh, A., Ramakrishnan, G., Chakrabarti, S.: Collective annotation of Wikipedia entities in web text. In: ACM KDD, pp. 457–466 (2009)
10. Medelyan, O., Milne, D., Legg, C., Witten, I.H.: Mining meaning from Wikipedia. Int. J. Hum.-Comput. Stud. 67(9), 716–754 (2009)
11. Milne, D., Witten, I.H.: An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In: AAAI Workshop on Wikipedia and Artificial Intelligence (2008)
12. Phan, X.H., Nguyen, L.M., Houriguchi, S.: Learning to Classify Short and Sparse Text & Web with Hiddent Topics from Large-scale Data Collections. In: WWW, pp. 91–100 (2008)
13. Sahami, M., Heilman, T.D.: A web-based kernel function for measuring the similarity of short text snippets. In: WWW, pp. 377–386 (2006)
14. Schlimmer, J.C., Graner, R.H.: Beyond Incremental Processing: Tracking Concept Drift. In: AAAI, pp. 502–507 (1986)
15. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. 34, 1–47 (2002)
16. Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., Demirbas, M.: Short text classification in twitter to improve information filtering. In: ACM SIGIR, pp. 841–842 (2010)
17. Strube, M., Ponzetto, S.P.: WikiRelate! Computing Semantic Relatedness Using Wikipedia. In: AAAI, pp. 1419–1424 (2006)
18. Sun, X., Haofen, W., Yong, Y.: Towards effective short text deep classification. In: ACM SIGIR, pp. 1143–1144 (2011)
19. Zelikovitz, S., Hirsh, H.: Improving short-text classification using unlabeled data for classification problems. In: ICML, pp. 1191–1198 (2000)
20. Zelikovitz, S., Marquez, F.: Transductive Learning for Short-Text Classification problems using Latent Semantic Indexing. IJPRAI 19(2), 146–163 (2005)