Representation Models for Text Classification: a comparative analysis over three Web document types

George Giannakopoulos[◇], Petra Mavridi[$],
Georgios Paliouras[◇], George Papadakis[$,♯], Konstantinos Tserpes[$]

[♯] L3S Research Center, Hanover, Germany papadakis@L3S.de
[◇] SKEL - NCSR Demokritos, Athens, Greece {ggianna, paliourg}@iit.demokritos.gr
[$] National Technical University of Athens, Greece {pmavridi, gpapadis, tserpes}@mail.ntua.gr

ABSTRACT

Text classification constitutes a popular task in Web research with various applications that range from spam filtering to sentiment analysis. To address it, patterns of co-occurring words or characters are typically extracted from the textual content of Web documents. However, not all documents are of the same quality; for example, the curated content of news articles usually entails lower levels of noise than the user-generated content of the blog posts and the other Social Media.

In this paper, we provide some insight and a preliminary study on a tripartite categorization of Web documents, based on inherent document characteristics. We claim and support that each category calls for different classification settings with respect to the representation model. We verify this claim experimentally, by showing that topic classification on these different document types offers very different results per type. In addition, we consider a novel approach that improves the performance of topic classification across all types of Web documents: namely the n-gram graphs. This model goes beyond the established bag-of-words one, representing each document as a graph. Individual graphs can be combined into a class graph and graph similarities are then employed to position and classify documents into the vector space. Accuracy is increased due to the contextual information that is encapsulated in the edges of the n-gram graphs; efficiency, on the other hand, is boosted by reducing the feature space to a limited set of dimensions that depend on the number of classes, rather than the size of the vocabulary. Our experimental study over three large-scale, real-world data sets validates the higher performance of n-gram graphs in all three domains of Web documents.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*

WIMS'12, June 13-15, 2012 Craiova, Romania.

General Terms

Algorithms, Experimentation

Keywords

Text classification, N-gram graphs, Web document types

1. INTRODUCTION

Text classification (**TC**), also known as *text categorization*, is the task of automatically detecting one or more predefined categories that are relevant to a specific document [30, 31]. This process is typically carried out with the help of supervised machine learning techniques; a classification algorithm is trained over a corpus of labeled documents in order to capture the most distinguishing category patterns that will be used to classify the new, unlabeled instances. TC constitutes a popular research topic, due to its applications in all kinds of Web documents: filtering spam out of e-mails [22], categorizing Web pages hierarchically [7] and analyzing the sentiment of Social Media content [27]. Therefore, its performance is critical for a wide range of tasks on the Web.

At the core of TC methods lies the representation model for (Web) documents, which defines the features that form the basis for applying classification techniques. Two are the dominant models that are typically employed in this context: the *term vector* and the *n-grams model*, collectively called *bag-of-tokens models* [20, 30]. The former represents documents - and categories - as a bag of (frequent) words, and the latter as a bag of (frequent) character or word sequences. Basically, they associate individual documents as well as classes with frequent, discriminative tokens or characters and categorization is based on the similarity between them.

The performance of these models depends heavily on the inherent characteristics of the document collection at hand. In fact, their effectiveness is degraded by semantically incorrect (or incomprehensible) phrases and by spelling, syntactical and grammatical mistakes, as these characteristics introduce noise to the information conveyed by a document. However, not all types of documents convey the same levels of noise. In the case of Web pages like news article, noise is typically very low, guaranteeing high classification accuracy. In contrast, the controversial, user-generated content of Web 2.0 and Social Media (e.g., messages on Twitter[1]

[1] `http://twitter.com`

and comments on Youtube[2] videos) involves many intricacies that affect not only the accuracy but also the efficiency of TC [1]. More specifically, it poses the following serious challenges to the functionality of traditional representation models [11, 13, 24]:

(C1) Multilinguality. Most representation models are *language specific*: to ensure high performance, they fine-tune their functionality to the language at hand. This is typically done with pre-processing techniques, such as lemmatization, stemming and word sense disambiguation with the help of dictionaries (e.g., WordNet[3]). Social Media posts can be in any language, but they typically lack any metadata that denotes it.

(C2) Sparsity. Social Media content solely comprises free-form text that is rather short in length, especially when compared to traditional Web documents, like Web pages. Due to size limitations, individual messages typically consist of a few words, thus involving little extra information that can be used as evidence for identifying the corresponding category.

(C3) Noise. Social Media posts are particularly noisy, due to their casual, real-time nature and their minimal curation. For example, users frequently participate in chats, posting their messages as quickly as possible, without verifying their grammatical or spelling correctness; incomprehensible messages can be simply corrected by a subsequent post.

(C4) Evolving, non-standard vocabulary. A large part of the activity in Social Media pertains to informal communication between friends, who typically employ a casual "communication protocol" (e.g., slang words and dialects) [8]. The limited size of their messages also urges them to shorten words into neologisms that bear little similarities to the original ones (e.g., "gr8" instead of "great").

In this paper, we start by providing a preliminary study of the endogenous characteristics of Web documents with respect to four dimensions. Based on it, we introduce three main types of Web Documents: the *curated*, the *semi-curated* and the *raw* ones. We claim and support that the selected document types cause variation in the performance of text classification systems, not only in terms of effectiveness, but also of efficiency. We provide empirical evidence for our claim by examining three large-scale, real-world data sets — one for each document type.

Our experimental study also highlights the inadequacy of the established representation models in handling the demanding content of Social Media. To deal with them, we apply a novel, efficient and language-neutral representation method that is robust to noise: the *n-gram graphs*. It goes beyond the plain bag-of-tokens models by representing individual documents and entire categories as graphs: their nodes correspond to specific n-grams, with their weighted edges denoting how close the adjacent n-grams are found on average. In this way, it adds contextual information to the n-grams model, thus achieving higher accuracy. It also improves the time efficiency of learning, by addressing successfully the problem of the "dimensionality curse": documents are classified according to a limited set of graph similarity metrics, with the overall number of features depending on the number of classes, instead of the vocabulary size. We compare n-gram graphs with the established representation models over the three real-world data sets of our experimental study. The outcomes demonstrate the significantly higher performance of n-gram graphs, not just for Social Media content, but across all types of Web documents.

On the whole, the main contributions of this paper are the following:

1. We categorize Web documents into three main types on the basis of their endogenous information. We analyze their inherent characteristics and demonstrate empirically that the type of a document affects the performance of TC, especially for the traditional representation models (i.e., bag-of-tokens).
2. We explain how the n-gram graphs can be employed as a representation model for TC and elaborate on its advantages over the existing models with respect to both accuracy and time efficiency.
3. We conduct an analytical experimental study with three large-scale, real-world data sets, one for each type of Web documents. Its outcomes demonstrate that the representation models behave differently in each case, with the n-gram graphs offering top performance across all types.

The rest of the paper is structured as follows: in Section 2, we formally define the problem we study and, in Section 3, we analyze the n-gram graphs model, comparing it with the traditional, bag-of-tokens ones. Section 4 defines the three types of Web documents, while Section 5 presents our experimental evaluation. In Section 6, we elaborate on existing work, and we conclude the paper in Section 7, along with directions for future work.

2. PROBLEM DEFINITION

Text classification has been extensively studied over the years, either as a stand-alone research domain [30], or as part of the broader field of *text mining* [3]. Related literature covers two main sub-problems: the *single-label* and the *multi-label TC*; the former involves disjoint categories (i.e., each document is assigned to a single class), whereas the latter allows for overlapping categories, associating each document with a multitude of classes. Both flavors of TC have evolved to cover a variety of different classification settings, ranging from document categorization [20] to genre and author classification [33] and spam detection [22].

In the following, we exclusively consider the single-label version of TC, since it is more general than multi-label TC: the latter can be split into several binary (i.e., single-label) classification problems, but the contrary is not possible [30]. Thus, any method that successfully tackles the former, is expected to exhibit a high performance for the latter, as well. In addition, most literature revolves around single-label classification, since it lies at the core of the main TC applications. A prominent example is text filtering (i.e., the process of distinguishing documents into relevant and irrelevant ones), with spam filtering probably constituting its most popular instantiation [30].

Among the applications of TC, we consider *topic classification* as our use case for examining the qualitative and

[2] `http://www.youtube.com`
[3] `http://wordnet.princeton.edu`

quantitative differences between our document representation models. This is actually the task of categorizing a given set of documents into thematic categories and is crucial in many applications of the Web, ranging from news services to blogs and Social Media [24].

More formally, the problem we are tackling in this work is defined as follows [30]:

Definition 1. *Given a corpus of documents $\mathcal{D}$, a set of topics $\mathcal{T}$, and a set $D_{tr} \subset \mathcal{D}$ of training pairs $D_{tr} = \{< d_i, t_i >, d_i \in \mathcal{D}, t_i \in \mathcal{T}\}$, we seek a function $f : \mathcal{D} \rightarrow \mathcal{T}$, that minimizes the size $|E|$ of the set of false pairs (i.e., errors): $E = \{< d_i, t_i >: d_i \in \mathcal{D}, f(d_i) \neq t_i\}$.*

Given that we only have a subset of the full corpus as a training set, we may fail to find the optimal function f, and we are rather looking for the best approximation to it. In this context, we additionally consider the following questions with respect to topic classification: *What is a good representation of Web documents? How can we use this good representation in conjunction with existing machine learning techniques? Can we develop a representation model that faces the computational and sparsity challenge of the user-generated content that is available on the Web? Is the accuracy and the speed of classification the same across different quality types of documents (e.g., the curated content of news articles and the noisy content of Social Media)?* The outcomes of our study are expected to be directly applicable to other cases of text classification, as well.

Note that the above definition exclusively relies on *endogenous* (i.e., content-based) information, assuming that the individual input documents solely consist of their textual content. Thus, it disregards any exogenous information, such as related Web resources and special-purpose metadata like publication date, which are often introduced in TC to enhance its performance. The reason is that such information may involve high extraction cost and, most importantly, it constitutes an application-dependent parameter [30]. Given that we do not aim at optimizing the performance of a specific application, we do not consider such information in our analysis. Instead, our goal is to examine the effect of the aforementioned challenges - C1 to C4 - in the performance of document representation models, identifying the one that adds more value to the process of capturing textual patterns.

3. DOCUMENT REPRESENTATION

The representation models for topic classification can be classified in two broad categories: those based exclusively on the endogenous information of the given corpus (i.e., *content-based models*), and those exploiting additional, external information in order to acquire more contextual information (i.e., *context-aware models*). As explained above, the latter are application-dependent and lie out of the scope of this work. Thus, in the following, we exclusively consider content-based representation models[4]. We overview the traditional, bag-of-tokens representations and then introduce the n-gram graphs as a richer alternative. A qualitative analysis of their advantages and disadvantages follows, highlighting the characteristics of n-gram graphs that account for their potential.

[4]It is worth noting at this point that contextual information typically come in the form of text, and content-based representation models are typically applied on them, as well.

3.1 Term Vector Model

This method constitutes the dominant Information Retrieval technique for detecting the relevant documents to a keyword query [26]. In the context of TC, it is employed as follows: given a collection of documents $\mathcal{D}$, it aggregates the set of distinct terms (i.e., words) $\mathcal{W}$ that are contained in it. Each document $d_i \in \mathcal{D}$ is then represented as a vector $\mathbf{v_{d_i}} = (v_1, v_2, \ldots, v_{|\mathcal{W}|})$ of size $|\mathcal{W}|$ with its j-th dimension v_j quantifying the information that the j-th term $w_j \in \mathcal{W}$ conveys for d_i. The same representation may apply to topics (i.e., categories), as well: their vectors comprise the terms that have been aggregated from the documents they entail[5].

The term information in each dimension can come in any of the following forms: (i) a binary value indicating the existence (or absence) of a term in the corresponding document, (ii) an integer value indicating the number of occurrences of a term in a document (i.e., *Term Frequency*), and (iii) a *Term Frequency-Inverse Document Frequency* (TF-IDF) value (cf. [29]). The last alternative takes into account both the number of occurrences of a term in a document and its overall frequency in the entire corpus in order to reduce the impact of particularly common words (i.e., stop words). This typically results in higher performance, which explains why the TF-IDF weights are usually preferred over the other two choices. In the following, we consider only this variation of the term vector model.

3.2 N-Grams Model

The model comes in two forms: the *character n-grams* model, which relies on sequences of distinctive, frequent letters, and the *word n-grams model*, which relies on sequences of distinctive, frequent words. The former outperform the latter in several application areas, such as spam filtering [21], authorship attribution [9] and utterance recognition [35]. Given also that in the settings we are considering it is usually difficult to identify words or even tokens[6], we exclusively consider character n-grams in the following.

The set of *character n-grams* of a word or sentence comprises all substrings of length n of the original text. A document d_i is, thus, represented by a vector whose $j-th$ dimension encapsulates the information conveyed by the $j-th$ n-gram for d_i. Unlike the term vector model, the frequency of an n-gram is commonly used to quantify this information. Similarly, a topic t_i can be modeled as a vector that comprises the aggregate frequency of the n-grams contained in its documents (or the corresponding centroid). Typical values for n are 2 (*bigrams*), 3 (*trigrams*) and 4 (*four-grams*). For example, the phrase "home_phone" consists of the following trigrams: {hom, ome, me_, _ph, pho, hon, one}.

3.3 N-Gram Graphs Model

The *n-gram graphs model* was first used in [15] as a summary evaluation method. The rationale behind it is the idea that the bag model of character n-grams disregards the order of characters' appearance in the original text, thus missing valuable information. As a result, words or documents

[5]Note that an alternative is to extract the centroid of the term vectors of individual documents to form a class vector.

[6]Tokenization is a rather naive approach for most (western) languages, as tokens are typically delimited by whitespace. This does not hold, however, for such languages as Chinese, where multiple words can be concatenated in a single token that actually corresponds to a sentence.

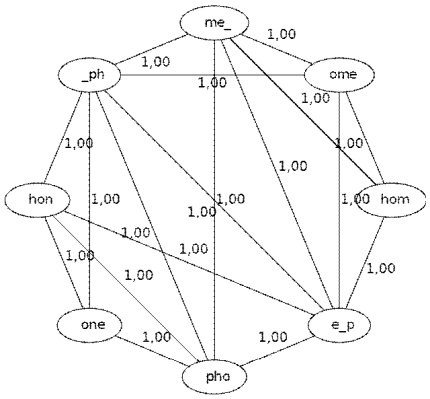


Figure 1: Tri-gram graph of "home_phone" string.

with different character sequences end up having identical or highly similar representations. For instance, the words "wiki" and "kiwi" have the same bigrams representation, although their meaning is totally different.

To overcome this problem, the n-gram graphs model associates neighboring pairs of n-grams with edges that denote their frequency of co-occurrence. An exemplary tri-gram graph, derived from the phrase "home_phone", is illustrated in Figure 1. Apparently, it conveys more information than the trigrams representation of the example in Section 3.2.

More formally, an n-gram graph is defined as follows [15]:

Definition 2 (N-Gram graph). *An n-gram graph is an undirected graph $G = \{V^G, E^G, W\}$, where V^G is the set of vertices that are labeled by the corresponding n-grams, E^G is the set of edges that are labeled by the concatenation of the labels of the adjacent vertices (in alphabetic order), and W is a function that assigns a weight to every edge.*

An n-gram graph is characterized by three parameters [15]: (i) the minimum n-gram rank $L_{\min}$, (ii) the maximum n-gram rank L_{MAX}, and (ii) the maximum neighborhood distance D_{win} [15]. Very low values of $L_{\min}$ and L_{MAX} (e.g., 1 or 2) are related to the alphabet and syllables of a language (possible combinations of characters). Higher values allow us to describe possible words or word subsequences, or even two-word substrings, providing more information than mere syllables. However, very high values induce noise (i.e., useless patterns attributed to change). In the following, we exclusively consider the configuration of $L_{\min} = L_{\text{MAX}} = D_{\text{win}} = n$ for values within previously studied limits ($n \in \{2, 3, 4\}$), which were theoretically shown and experimentally verified to provide good enough information, while limiting the presence of noise [15].

To represent a document d_i, we create a **document graph** G_{d_i} by running a window of size D_{win} over its textual content in order to analyze it into overlapping character n-grams. Any two n-grams that are found within the same window are connected with an edge $e^G_{d_i} \in E^G_{d_i}$, whose weight denotes their frequency of co-occurrence in the document. The document is, thus, transformed into a graph that — in addition to its n-grams — captures the contextual information of their co-occurrence.

This representation can also be employed for an entire topic (i.e., set of documents). In this case, however, the graph is derived from the merge of the individual document graphs, similarly to the concept of a centroid vector. The graph models of the topic's documents are merged into a single **class graph** through the *update operator* [16] as follows. Given a collection of documents $\mathcal{D}$, an initially empty graph $G_{\mathcal{D}}$ is built; the $i-th$ document $d_i \in \mathcal{D}$ is then transformed into the document graph G_{d_i} that is merged with $G_{\mathcal{D}}$ to form a new graph $G^u_{\mathcal{D}}$ with the following properties: its edges (nodes) comprise the union of the edges (nodes) of the individual graphs, and its weights are adjusted so that they converge to the mean value of the respective weights. More formally: $G^u_{\mathcal{D}} = (E^u, V^u, W^{u_i})$, where $E^u = E^{G_{\mathcal{D}}} \cup E^{G_{d_i}}$, $V^u = V^{G_{\mathcal{D}}} \cup V^{G_{d_i}}$ and $W^{u_i}(e) = W^{G_{\mathcal{D}}}(e) + (W^{G_{d_i}}(e) - W^{G_{\mathcal{D}}}(e)) \times 1/i$, where the division by i ensures the incremental convergence to the overall average value [16]. The resulting class graph captures patterns common in the content of the entire topic, such as recurring and neighboring character sequences and digits.

The similarity between documents and topics is estimated through the closeness of their graph representations. The following graph similarity metrics are used in this work:

1. **Containment Similarity (CS)**, which expresses the proportion of edges of a graph G^i that are shared with a second graph G^j. Assuming that G is an n-gram graph, e is an n-gram graph edge and that for function $\mu(e, G)$ it stands that $\mu(e, G) = 1$, if and only if $e \in G$, and 0 otherwise, then:
$$\text{CS}(G^i, G^j) = \frac{\sum_{e \in G^i} \mu(e, G^j)}{\min(|G^i|, |G^j|)},$$
where $|G|$ denotes the number of edges of graph G (i.e., the *size of the n-gram graph*).

2. **Size Similarity (SS)**, which denotes the ratio of sizes of two graphs:
$$\text{SS}(G^i, G^j) = \frac{\min(|G^i|, |G^j|)}{\max(|G^i|, |G^j|)}.$$

3. **Value Similarity (VS)**, which indicates how many of the edges contained in graph G^i are contained in graph G^j, as well, considering also the weights of the matching edges. In this measure, each matching edge e having a weight $W^i(e)$ in graph G^i contributes $\text{VR}(e)/\max(|G^i|, |G^j|)$ to the sum, where VR(e) (i.e., value ratio) is a symmetric, scaling factor that is defined as $\text{VR}(e) = \frac{\min(w^i_e, w^j_e)}{\max(w^i_e, w^j_e)}$, thus taking values in the interval $[0, 1]$. Non-matching edges do not contribute to VS: $w^i_e = 0$ for an edge $e \notin G^i$. Plugging all these measures together, we have:
$$\text{VS}(G^i, G^j) = \frac{\sum_{e \in G^i} \frac{\min(w^i_e, w^j_e)}{\max(w^i_e, w^j_e)}}{\max(|G^i|, |G^j|)}.$$
VS converges to its maximum value $\text{VS}_{max} = 1$ for graphs that share both the edges and the corresponding weights, with VS_{max} indicating perfect match between the compared graphs.

An important, derived measure is the **Normalized Value Similarity (NVS)**, which is computed as follows:
$$\text{NVS}(G^i, G^j) = \frac{\text{VS}(G^i, G^j)}{\text{SS}(G^i, G^j)}.$$

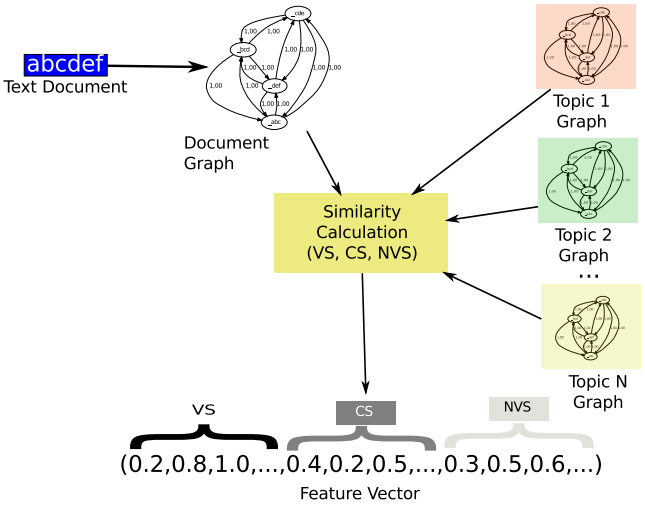


Figure 2: Extracting the feature vector from the n-gram graphs model.

The NVS enhances VS by disregarding the relative size of the compared graphs.

In essence, the containment similarity between two n-gram graphs implies co-occurrence of similar substrings in the corresponding texts. It is related, as a notion, to the cosine similarity in the vector space of the n-grams model over binary values; however, CS considers the co-occurrence of pairs of n-grams (i.e., edges), instead of the co-occurrence of individual n-grams. (Normalized) Value similarity, on the other hand, takes into account the frequency of co-occurrence of n-grams, thus being analogous to the cosine similarity between frequency-based vectors of the n-grams model. Once again, though, (N)VS functions on pairs of n-grams, instead of individual ones.

In this work, to classify a document using the n-gram graphs model we first calculate the class graphs from the training instances. Each unlabeled document is then positioned into a vector space, as follows:

- The document is represented as a graph (i.e., document graph).
- For every class, we compare the document graph with the corresponding class graph to derive the similarities that comprise the feature vector. In more detail, we extract 3 features from each comparison, one for each of the similarity measures CS, VS and NVS. The result, is that we get 3 similarity features per class, for our document.
- Given N class graphs, the resulting feature vector contains $3 \times N$ similarity-based features.

The overview of this embedding process is illustrated in Figure 2.

3.4 Qualitative analysis

Having outlined the functionality of the main TC representation models, this section elaborates on the qualitative aspects of their performance, explaining how it is affected by the settings we are considering (i.e., the four challenges of Section 1 - C1 to C4).

Starting with the term vector model, it is worth noting that its most critical step is the identification of the distinct words among a collection of documents. The reason is that the same word may appear in different forms (e.g., a verb may appear as a gerund or in some other inflected form). In addition, the same word might have a different meaning, depending on its context. To ensure high performance, Information Retrieval preprocessing methods are usually employed to group together different manifestations of the same word or meaning. In this way, the number of dimensions in the feature space is restricted, allowing for a more efficient classification. In domains with highly diverse vocabulary (i.e., C4), feature selection can be critical for the effectiveness (i.e., accuracy) of classification, as well, since many features may constitute noise (see [12] for a related study of feature selection methods).

A common preprocessing technique is *stemming* [26];, it reduces the inflected or derived words to their root form, usually by removing their suffix (e.g., it removes the plural "s" from the nouns in English). *Lemmatization* improves on this process by taking into account the context of a word - or even grammar information - in order to match it to a lemma. Both these methods require language-dependent knowledge to function, thus having limited effectiveness in multilingual settings (i.e., C1). The performance of the term vector model can be significantly degraded by spelling mistakes (i.e., C3), as well, which hinder the process of detecting and clustering together the different appearances of the same word. This leads to an extensively larger feature space (i.e., lower efficiency) as well as to lower effectiveness, due to noise.

The character n-grams model improves on both disadvantages of the term vector one, constituting a language-neutral technique that is highly robust to noise (especially with respect to spelling mistakes). Thus, it successfully deals with challenges C1 and C3, respectively. As mentioned above, however, its performance is restricted by the fact that it completely ignores the sequence of n-grams inside a phrase.

A serious drawback, common to both models, is the *curse of dimensionality*: the number of features that they entail is usually very high - depending, of course, on the size of the corpus (i.e., number of documents). In absolute numbers, it is higher for the n-grams than for the term vector model (for the same corpus), increasing with the increase of n; the reason is that sub-word tokens are typically more frequent than whole words, and the larger the values of n is, the higher is the number of possible character combinations. This situation is particularly aggravated in the context of a highly diverse vocabulary: the more heterogeneous a document collection is - either with respect to the languages it comprises (i.e., C1) or the regional variations used by its authors (i.e., C4) - the higher is the number of features that these methods take into account. A common practice for restricting the impact of this problem is to set a threshold on the minimum frequency of the terms that are considered as features. This practice, however, is a mere heuristic procedure, whose performance is application-dependent.

In contrast, the n-gram graphs method involves a limited feature space, whose dimensions solely depend on the number of distinct classes. In addition, our model makes no assumptions on the underlying language, thus being able to handle the multilingual, user-generated content that is available on the Web (i.e., C4). It also allows for fuzzy matching

Document Type	Document Size	Vocabulary Size	Special Notation	Noise
Curated	Long	Formal	No	Negligible
Semi-curated	Average	Colloquial	Yes	Low
Raw	Short	Slang	Yes	High

Table 1: Content-based criteria for determining the quality type of a Web document.

and substring matching, which constitutes a functionality of high importance in open domains, like the content of Social Media (i.e., C3). Its only drawback is the time that is needed in order to construct a class-representative graph and to compute the graph similarities. As demonstrated in [15], the time complexity of these processes depends both on the size of n and the size of the input document collection $|D|$.

Last but not least, the n-gram graphs model fundamentally differs from the other two in the way it tackles C2: it ameliorates the effect of sparsity by encapsulating contextual information in its edges, whereas the bag-of-tokens models make no provision for this challenge, relying exclusively on the inadequate features extracted from the sparse content.

4. WEB DOCUMENT TYPES

The above analysis provided hints as to the traits of Web documents that affect the effectiveness as well as the efficiency of their representation models. In what follows, we elaborate on these traits, we explain how they affect the performance of topic classification and we analyze how they can be used to distinguish among three main types of Web documents. Note that these types are not intended for genre classification, as we solely aim at explaining the inherently different quality of their content and its impact on classification. Note also that the problem we are tackling in this work completely disregards exogenous meta-data. Thus, our analysis takes into account only factors that can be derived directly from the textual content of a document.

We consider the following — qualitative and quantitative — criteria as the most critical ones for the performance of topic classification:

1. *Document Size* expresses the length of a document with respect to the number of characters (or tokens) it comprises. The higher its value is, the higher is the number of (possible) features for the bag-of-tokens models. This criterion is directly related to challenge C2, as the shorter a document is, the more sparse is the information it comprises.

2. *Vocabulary Size* denotes the diversity of the words and phrases that are employed in a document. Higher diversity corresponds to a higher number of tokens that can be possibly used as features. As a result, Vocabulary Size is increased by multilinguality (i.e., C1) and by the evolving, non-standard expressions used in Social Media content (i.e., C4). To express it in a comprehensible way, we associate it with the type of language that a document is written in. We acknowledge the following language types: (i) *formal*, (ii) *colloquial*, and (iii) *slang*. The first type generally involves controllable levels of diversity, due to the standard expressions it entails; it is the language that a journalist typically employs to record facts in a news article. Slang language involves the largest vocabulary size, due to the informal, rich in neologisms language that is employed when communicating through instant messages. Colloquial language lies in the middle of these two extremes.

3. *Special Notation* denotes whether a document contains non-verbal expressions that actually constitute hyperlinks to some Web resource: links to Web pages, links to multimedia content (i.e., videos or images), or even links to users (e.g., the @*username* notation used in Twitter). This kind of special notation is typically employed to enhance the effectiveness of topic classification by introducing valuable meta-data information (e.g., [24]). As a content-based feature, however, it may add noise and, thus, it is relevant to challenge C3.

4. *Noise* is directly related to challenge C3, reflecting the level of spelling mistakes as well as of grammatically, syntactically and semantically incorrect phrases in the text of a document. Such errors distort its actual meaning, thus hindering the detection of word or n-gram patterns. Low noise ensures, therefore, high reliability of the content-based features, and vice versa.

We argue that these criteria capture main characteristics of the major kinds of contemporary Web documents: typical (static) web pages, discussion fora, blogs as well as Social Media. In fact, we distinguish three types of Web documents with the help of these criteria:

1. the *curated* documents, which entail large documents of pure text (i.e., without notations) with standard, formal vocabulary and low levels of noise,

2. the *semi-curated* documents, which are shorter in size, involve more noise, a slightly larger vocabulary, and plenty of hyperlinks, and

3. the *raw* documents, which are rather telegraphic, noisy and rich in special notation.

The characteristics of these Web document types are outlined in Table 1. We further elaborate on them in the following, analyzing the implications they convey in the process of topic classification.

4.1 Curated Documents

This type comprises such documents as news articles and scientific publications. The text is adequately long to pose well-described questions, to provide argumentation or to cover a topic; sparsity (i.e., C2), therefore, is not an issue. Spelling mistakes are rather rare and the writing is correct - both with respect to grammar and syntactic rules -, since it has been edited or peer-reviewed; thus, it does not suffer from C3. Its language is eloquent and the text itself is focused and clear, lacking any neologisms and non-standard vocabulary (i.e., absence of C4). The content is multilingual, but its actual language is commonly known a priori. Words are, thus, easily grouped into features through lemmatization and stemming, overcoming the challenge of C1.

In combination with the character n-grams model, it leads to a feature space of high dimensionality that exhibits high effectiveness. The negligible level of noise, though, does not provide it with a significant advantage over the term vector model. Given that the latter has a lower dimensionality — thus being more efficient — the character n-grams method

may constitute a sub-optimal choice for the classification of curated documents. The n-gram graphs approach is expected to outperform both of these models in terms of effectiveness, due to the contextual information that is encapsulated in its edges; even term collocations may be implied by the neighborhood of character n-grams. This method also exhibits the (probably) highest classification efficiency, due to the low number of features. The large size of the documents, however, results in large document and topic graphs, which involve considerable computational effort.

4.2 Semi-Curated Documents

This type of documents entails forum posts, text in wikis, e-mails, and personal blog posts. Their content is multilingual (i.e., C1), minimally edited by its author and comprises few paragraphs (i.e., C2), which are - nevertheless - long enough to analyze personal thoughts or to act as written dialogue parts. Neologisms, informal language and hyperlinks form part of its content (i.e. C4), with spelling mistakes and wrong sentences being relatively common (i.e., C3). On the whole, it involves all challenges of Section 1, though at a significantly lesser extent than Social Media content.

The term vector model is expected to have a large feature space, due to the large size of documents and the relatively rich vocabulary. The presence of noise is expected to have a significant impact on its effectiveness. Higher accuracy is, thus, achieved in combination with the the character n-grams and the n-gram graphs models, which are robust to noise. In the n-gram graphs case, though, the size of the documents may pose a serious computational cost for building and comparing the document and the topic graphs.

4.3 Raw Documents

This type refers to such documents as Facebook[7] status updates, YouTube comments, messages in Twitter (also termed *tweets*) and short posts in any Web 2.0 platform. A basic trait of these documents is that they are meant to be self-contained, conveying their message through a text of minimal size (i.e., C2); for instance, the messages are often meant to be the answer to questions like "What is new?", "What are you thinking?", "What is happening?". They can also comprise brief comments that simply convey an opinion or sentiment. Their authors typically use the full range of internet neologisms, abbreviations, emoticons and all other similar language constructs (i.e., C4). The grammar of the text is usually of minimal interest and it is not rare for non-fluent users to post messages, sometimes using a mixture of languages (i.e., C3). Their content is multilingual (i.e., C1) with high levels of geographic lexical variations, as well (i.e., C4); for instance, Twitter users from northern California write "koo" instead of cool, while the same word in southern California is mentioned as "coo" [8]. In summary, raw documents contain short, unedited, and noisy texts, abundant in special notations, which convey contextual information that may be essential to understand their meaning.

The very high levels of noise are expected to pose a significant barrier to the effectiveness of the term vector model. The character n-grams and n-gram graph models are expected to perform significantly better, due to their tolerance to noise. Furthermore, the efficiency of the n-gram graphs model is expected to improve in this context: the limited size of the raw documents entails minimal computational effort for the creation and the comparison of their graph representations.

[7] `http://www.facebook.com/`

Data Set	Class Label	Documents	Distribution
$\mathbf{D_{reuters}}$	ECAT	13,768	8.00%
	MCAT	41,523	24.13%
	CCAT	45,382	26.37%
	GCAT	71,442	41.51%
$\mathbf{D_{blogs}}$	Current Affairs	3,288	4.51%
	Entertainment	3,753	5.15%
	Blog	3,825	5.25%
	Work	4,095	5.62%
	Life	4,631	6.35%
	Personal	5,003	6.86%
	Politics	6,738	9.24%
	Music	8,295	11.38%
	News	12,971	17.79%
	Votes	20,320	27.87%
$\mathbf{D_{twitter}}$	#quote	99,385	2.51%
	#fact	157,959	3.98%
	#followfriday	220,155	5.55%
	#news	258,080	6.51%
	#musicmonday	307,322	7.75%
	#iranelection	320,310	8.08%
	#tcot	363,739	9.17%
	#ff	425,715	10.73%
	#jobs	866,752	21.85%
	#fb	947,058	23.88%

Table 2: Data set class distribution.

On the whole, we can argue that curated and raw documents define the two extremes of Web document quality with respect to morphology. The former involves large texts, where spelling mistakes and non-standard expressions are the exception, while the latter entails short texts with non-standard, slang expressions and a considerable portion of special notation. Semi-curated documents lie in the middle of these two extremes, slightly closer, though, to the curated ones: they involve middle-sized texts but with significantly more noise, non-standard expressions and special notation. We provide experimental evidence for these patterns in Section 5.1.

5. EXPERIMENTAL EVALUATION

The goal of this section is threefold: (i) to provide quantitative evidence for the above Web document types (Section 5.1), (ii) to illustrate whether the document type influences the performance of topic classification, and (iii) to provide an analytic comparison of the performance of the various representation models presented above (Section 5.2). Particular attention is paid to the trade-off between effectiveness and time efficiency, highlighting the balance between them that is achieved by every combination of representation model and document type.

In the following, we first present the real-world data sets we employ in our experimental study and provide empirical support to the differences between document types of Section 4. We then present the setup of our experiments, we elaborate on their outcomes with respect to effectiveness and, finally, on their outcomes with respect to efficiency.

5.1 Data Sets

To evaluate our document categorization in real settings, we considered three large-scale, real-world data sets — one for each type. They are analyzed individually in the following paragraphs.

Curated Documents. As representative for this type of

documents, we selected the Reuters RCV2 corpus[8], which has been widely used in the literature (e.g., [2, 5]). It constitutes a multilingual collection of news articles, dating from the time period between August 1996 and August 1997. In total, it contains over 480,000 articles, written in 13 different languages. For our analysis, we considered a subset of this collection, comprising 172,115 articles that are written in four languages: the German (16,888 documents), the Spanish (9,747 documents), the Italian (7,598 documents), and the French (7,490 documents). The news articles of RCV2 are categorized along a class hierarchy of 104 overlapping topics. In our experiments, we considered only the top four categories, which are non-overlapping, thus allowing for single-label TC. The distribution of documents among them is presented in Table 2. This data collection is denoted by $\mathbf{D_{reuters}}$ in the rest of the paper.

Semi-Curated Documents. For this type of documents, we selected the collection of blog posts that was published in the context of the 3^{rd} workshop on the Weblogging Ecosystem in 2006[9]. It contains around 10 million documents, stemming from approximately 1 million different weblogs. They were posted on-line in the time period between July 7, 2005 and July 24, 2005. For our analysis, we considered the documents corresponding to the 10 largest categories. We removed those documents that belong to more than one of the considered categories, since we examine the single-label TC. This resulted in 72,919 blog posts, in total, whose class distribution is presented in Table 2. Unfortunately, there is no direct information on the languages it contains. In the following, this data set is symbolized as $\mathbf{D_{blogs}}$.

Raw Documents. This type of documents is represented in our analysis by Twitter posts. We used the same data set as in [37], which comprises 467 million tweets that have been posted by around 20 million users in the time interval between June, 2009 and December, 2009. To derive the topic categorization of the tweets, we relied on their *hashtags*[10]. Around 49 million of the tweets are marked with at least one hashtag. As in $D_{reuters}$, we considered the subset of these documents that belong to the 10 largest topics. We removed the tweets that are associated with multiple categories as well as the retweets, which constitute reproductions of older tweets, thus containing no original information. The resulting collection — represented by $\mathbf{D_{twitter}}$ in the rest of the paper — comprises almost 4 million tweets. Note that, following [13, 24], we removed all hashtags from the tweets of $D_{twitter}$, since they are likely to contain category information. Again, there is no straightforward information on the languages that this data set contains.

5.1.1 Analysis of Document Types

It is interesting to examine the technical characteristics of the above data sets with respect to the parameters and types that were defined in Section 4. An overview of the relevant features is presented in Table 3. We can see that there is a large difference in the average size of the documents of the individual data sets: the news articles of $D_{reuters}$ contain 1,200 characters, or 180 tokens, on average, while $D_{twitter}$ is very sparse (i.e., C2), containing documents of just 95 characters, or 14 tokens. As stated in Section 4, the semi-curated documents of D_{blogs} lie closer to $D_{twitter}$, comprising 1,100 characters, or 160 tokens, on average. On the whole, however, the overall sizes of the data sets with respect to characters — or tokens — are very close, thus allowing for a comparison on an equal basis.

	$\mathbf{D_{reuters}}$	$\mathbf{D_{blogs}}$	$\mathbf{D_{twitter}}$
Classes	4	10	10
Documents	172,115	72,919	3,966,475
Total Characters ($\times 10^8$)	2.07	0.80	3.78
Characters/Document	1,205.55	1,100.25	95.41
Total Tokens ($\times 10^6$)	30.92	11.69	57.08
Tokens/Document	179.66	160.31	14.39
Av. Token Length	5.68	5.74	5.73

Table 3: Characteristics of data sets.

Data Set	Kind of Token	Length	Instances	Distrib.
$\mathbf{D_{blogs}}$	URL	72.36	125,495	1.07%
	Regular Word	5.02	11,564,370	98.93%
$\mathbf{D_{twitter}}$	Mention	11.42	1,867,706	3.27%
	URL	21.73	2,161,499	3.79%
	HashTag	5.94	4,003,308	7.01%
	Regular Word	4.79	49,045,538	85.93%

Table 4: Analysis of the special notation tokens.

It is also remarkable that the tokens of $D_{reuters}$ are smaller — on average — than those of D_{blogs} and of $D_{twitter}$. This is apparently a paradox, since our categorization argues that raw documents contain abbreviations and neologisms (i.e., C4), which should be substantially shorter than the original words (e.g., "gr8" instead of "great"). To investigate this phenomenon, we analyzed the special notations that are contained in the semi-curated and raw texts of D_{blogs} and $D_{twitter}$, respectively. As illustrated in Table 4, 1% of all tokens in D_{blogs} actually pertains to URLs, which are rather large in size (72 characters, on average). Regular words, on the other hand, typically consist of just 5 characters, thus being shorter than those in $D_{reuters}$.

In the case of $D_{twitter}$, the relative amount of special notations is significantly larger. Almost 15% of all tokens constitute "non-verbal tokens": 3% of them refer to some Twitter user (i.e., mentions), almost 4% are URLs and 7% designate the topic(s) of the tweet (i.e., hashtags). The remaining, regular words have an average length of 4.8 characters, thus being smaller than those of $D_{reuters}$ by a whole character.

On the whole, our experimental analysis validates our arguments about the quantitative parameters of the document types of Section 4.

5.2 Topic Classification

5.2.1 Set-up

To thoroughly test the performance of the document representation models, we considered two inherently different classification algorithms, which are typically employed in the context of TC: the Naive Bayes Multinomial (**NBM**) and the Support Vector Machines (**SVM**). The former is a highly efficient method that classifies instances based on the conditional probabilities of their feature values; the latter constitutes a more elaborate approach that uses optimization techniques to identify the maximum margin decision hyperplane. It is the state-of-the-art for TC, but does not

[8] http://trec.nist.gov/data/reuters/reuters.html

[9] http://www.blogpulse.com/www2006-workshop/datashare-instructions.txt

[10] A hashtag in Twitter consists of the symbol #, followed by a series of concatenated words and/or alphanumerics (e.g., #wsdm2012).

	$\mathbf{D_{reuters}}$	$\mathbf{D_{blogs}}$	$\mathbf{D_{twitter}}$
Vector Model	76.67%	49.81%	59.96%
Bigrams	56.02%	56.19%	57.83%
Trigrams	64.33%	61.53%	64.54%
Four-grams	71.19%	63.51%	65.06%
Bigram Graphs	50.81%	49.57%	41.91%
Trigram Graphs	90.71%	62.33%	65.63%
Four-gram Graphs	**93.71%**	**64.94%**	**69.79%**

Table 5: Precision of Naive Bayes Multinomial.

	$\mathbf{D_{reuters}}$	$\mathbf{D_{blogs}}$	$\mathbf{D_{twitter}}$
Vector Model	91.63%	63.91%	63.35%
Bigrams	89.46%	59.34%	71.42%
Trigrams	93.87%	64.35%	78.44%
Four-grams	94.71%	65.17%	78.19%
Bigram Graphs	91.32%	65.41%	76.86%
Trigram Graphs	95.09%	73.55%	**83.06%**
Four-gram Graphs	**95.44%**	**76.31%**	79.60%

Table 6: Precision of Support Vector Machines.

scale well to large datasets and to feature spaces with high dimensions[11]. We compare these two algorithms on an equal basis with the aim of identifying the classification conditions under which it is possible to sacrifice the high effectiveness of SVM for the high efficiency of NBM.

To measure the effectiveness of classification algorithms, we employed 10-fold cross validation; in every iteration, we used 90% of the input corpus as the training set and the remaining 10% as the testing set. To train a classifier over the n-gram graphs model, we randomly selected half of the training instances of each topic in order to build the corresponding class graph. All class graphs were then compared with the entire training set, providing the similarities that built the classifier, as well as with the entire testing set in order to estimate the classification **accuracy**. This measure of effectiveness is defined as follows:

$$accuracy = \frac{true\_positives}{true\_positives + false\_positives},$$

where *true_positives* stands for the number of documents that were assigned to the correct topic, and $false\_positives$ denotes the number of documents associated with a wrong topic.

Our approach and experiments were fully implemented in Java, version 1.6. The functionality of the n-gram graphs was provided by the open source library of JInsect[12]. For the implementations of the classification algorithms, we used the version 3.6 of the open source library of Weka[13] [17]. In every case, we employed the default configuration of the algorithms, without fine-tuning any of the parameters. Note that to scale the SVM in the large dimension space of the bag-of-tokens model and the large number of instances of $D_{twitter}$, we employed the LIBLINEAR optimization technique [10] through its Weka API[14]. Given that LIBLINEAR employs linear kernels for training the SVM, it is directly comparable with the default configuration of SVM in Weka with linear kernels, which was employed in all other cases. All experiments were performed on a desktop machine with 8 cores of Intel i7, 16GB of RAM memory, running Linux (kernel version 2.6.38).

Note that for the term vector and the character n-grams model we did not employ any preprocessing technique, as the language of the documents is unknown (i.e., multilingual settings). In order to limit the feature space, we set a threshold of minimum frequency for each feature, discarding those not exceeding it. For $D_{reuters}$ and D_{blogs}, this threshold was equal to 1% of the size of the document collections, whereas for $D_{twitter}$ it was set to 0.1%, due to the substantially larger number of document it comprises. These limits resulted in relatively stable number of features for each representation model across all document types.

[11]For a comprehensive overview of the algorithms' functionality, see [36].

[12]`http://sourceforge.net/projects/jinsect`

[13]`http://www.cs.waikato.ac.nz/ml/weka`

[14]`https://github.com/bwaldvogel/liblinear-weka`

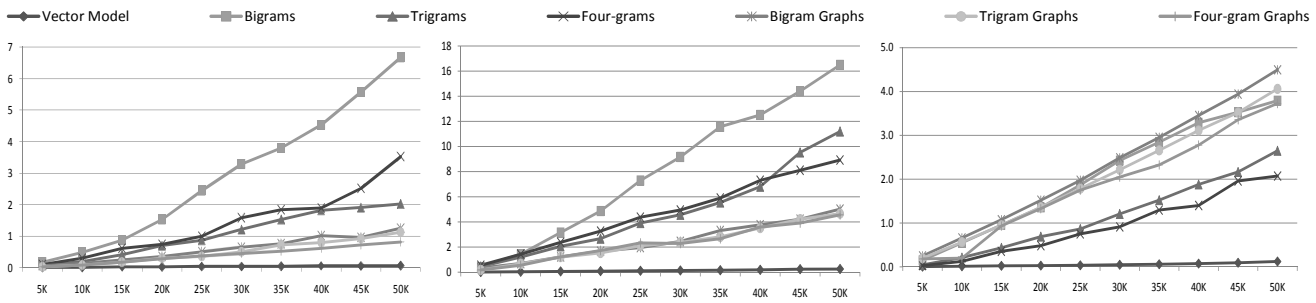
5.2.2 *Effectiveness Experiments*

The performance of the representation models over the NBM and the SVM classification algorithms are presented in Tables 5 and 6, respectively. Note that the unequal distribution of the classes in each data set results in different baseline values for accuracy. The *random classifier* (i.e., the classifier that assigns all instances to the largest topic) has an accuracy of 41.51% for $D_{reuters}$, 27.87% for D_{blogs} and 23.88% for $D_{twitter}$. Nevertheless, all combinations of representation models and classification algorithms perform significantly higher than the baseline.

Most importantly, though, we can notice that the numbers in Tables 5 and 6 follow several interesting patterns. First, we can see that the performance of the n-grams model increases with the increase of n in the case of $D_{reuters}$ and D_{blogs}, independently of the classification algorithm (i.e., four-grams achieve the highest accuracy in both datasets). In $D_{twitter}$, however, there is no clear difference between trigrams and four-grams, as the former have a slightly higher effectiveness than the latter for SVM and vice versa for NBM; these differences, though, are statistically insignificant and are probably related to the low level of noise in the case of the curated and semi-curated documents of the first two data sets; the higher portion of spelling mistakes in the raw documents, on the other hand, provides a significant boost to the performance of trigrams. As for the bigrams, they seem inadequate to capture distinguishing textual patterns, due to their limited length, thus having an extensively lower accuracy in most of the cases.

As expected, the same pattern applies to the n-gram graphs model, as well. The low levels of noise in (semi-)curated documents boosts the performance of four-gram graphs, while the noisy content of raw documents favors the trigram graphs: the latter outperforms the former for SVM and vice versa for NBM. It is worth noting at this point that the n-gram graphs models exhibit a consistently higher accuracy than the corresponding n-grams models for both classification algorithms, due to the additional, contextual information they encapsulate. The only exceptions to this rule are the bigram graph models, which have a consistently lower accuracy than the bigrams model, when combined with NBM. Thus, we can safely conclude that the n-gram graphs are more effective than the plain n-grams, independently of the documents' type. We note that in [15] one can find a method to calculate a near-optimal n size for n-gram graphs (applicable to character n-grams as well), but we avoided to use it due to its poor scalability over large corpora.

Regarding the relation between the term vector and the n-



(a) $D_{reuters}$ **(b) D_{blogs}** **(c) $D_{twitter}$**

Figure 3: Training time (in minutes) of SVM over various sample sizes per representation model and document type.

grams model, we can easily notice that, although the former extensively outperforms the latter for $D_{reuters}$, the situation is totally reversed for D_{blogs} and $D_{twitter}$. Note, though, that the deviation in the accuracy of these models is relatively low for D_{blogs}, but significantly higher for $D_{twitter}$. This behavior is in complete agreement with the theoretical expectations of Section 4, which argues that the low noise of the curated documents provides an advantage to the term vector model over the n-grams one.

The relation between the term vector and the n-gram graphs model follows a different pattern: for most combinations of classification methods and document types, the latter significantly outperforms the former. The only exceptions are the bigram graphs, especially when used in conjunction with NBM. This combination actually has the lowest performance across all cases; given that the accuracy of bigram graphs is substantially enhanced when combined with SVM, we can infer that NBM is inadequate for handling their contextual information. Nevertheless, we can safely deduce that the n-gram graphs model constitutes a more suitable representation for TC than the term vector one, regardless of the document type.

On the whole, we can conclude that the n-gram graphs model provides the highest effectiveness for all types of documents, independently of the classification algorithm. Their optimal configuration seems to depend on the document type. On the other hand, the difference in performance is such that allows use of any value between 3 and 4 for n. Regarding the classification algorithm to be used, SVM exploits the contextual information of n-gram graphs more effectively than NBM, but the latter provides a highly efficient, adequate alternative in the case of curated documents.

5.2.3 *Efficiency Experiments*

In this section, we examine the effect of representation models on the time efficiency of classification training. We consider two metrics: the number of features a model extracts from the training set, and the *training time* of a classifier (i.e., the wall-clock time - in minutes - that is required for training a classifier over a collection of documents that has been transformed into the given representation model). In general, the higher the dimensionality of the feature space is, the higher the training time is expected to be.

Table 7 presents the number of features each representation model extracts from each data set of our study. We can notice that the n-grams model employs the largest feature space in all cases, with higher values for n resulting in more dimensions. The term vector model entails significantly less features than the n-grams one in all cases, since - as explained in Section 3 - there is usually a larger variety of sub-words than words. The number of features the n-gram graphs models employs is lower by two orders of magnitude across all document types.

	$D_{reuters}$	D_{blogs}	$D_{twitter}$
Term Vector	1,742	1,560	1,263
Bigrams	2,129	2,113	2,758
Trigrams	9,327	7,381	7,609
Four-grams	17,891	14,211	12,659
Bigram Graphs	12	30	30
Trigram Graphs	12	30	30
Four-gram Graphs	12	30	30

Table 7: Features per representation and data set.

To measure the actual classification time that these feature sets entail, we used the LibLINEAR algorithm in its default configuration. For each representation model, we considered 10 sample sizes, from 5,000 up to 50,000 documents, with a step of 5,000. The instances of each sample were randomly selected from the final set of training instances of each model. The procedure was repeated 10 times and the average training times are presented in Figures 3(a) to (c). Note that we do not present the number of features involved by the bag-of-tokens models in each sample size, since - on average - it was pretty close to that over the entire data set, presented in Table 7.

Looking into these diagrams, we can easily notice that the vector space model requires the lowest training time (less than a minute) in all cases. This is probably because it results in a significantly lower number of outliers, which typically dominate the efficiency of the SVM. The performance of the n-gram graph models follows two patterns: first, it exhibits a strong correlation with the number of features (i.e., number of classes) that are involved in each case; for the $D_{reuters}$ that involves only 4 classes its training time is around 1 minute, rising to approximately 4 minutes in the case of 10 classes of the data sets D_{blogs} and $D_{twitter}$. Second, its training time is almost identical for the different sizes of n, for each sample size of a specific data set. In fact, there is just a slight increase in the efficiency with the increase of n; that is, the bigram graphs involve slightly higher time, followed by the trigram graphs and the four-gram ones. This is probably because the larger the size of n is, the more discriminative the features are and the lower the portion of outliers is. This applies to the n-grams model, as well, since bigrams require significantly higher training time than tri-

grams and four-grams, which exhibit almost identical levels of efficiency.

In general, the n-grams model appears to be the least efficient one, involving a training time that is two or more times higher than that of the n-gram graphs. Only in the case of $D_{twitter}$, they practically share identical levels of efficiency across most sample sizes. This is probably because of the the very small size of raw documents, which leads to very sparse n-grams representations that are processed quite rapidly. In contrast, the efficiency of the n-gram graphs model is not affected by the average document size, having a stable training time that depends exclusively on the number of involved classes. The most efficient model in call cases, though, is the term vector one, but its low effectiveness justifies its use only in the case of curated documents.

6. RELATED WORK

Questions commonly posed in the contemporary text classification literature refer to: (i) the representation of text instances, (ii) techniques for facing sparseness in the context of limited information (e.g., short text instances), and (iii) methods for integrating the specifics of a domain (e.g., social media, product reviews, e-mails).

Apart from the representation models of Section 3, document models based on "latent topics" have also been employed in topic classification. They represent documents and topics in the vector space that is defined by the latent topics of the input corpus; these topics are typically identified using methods like the Latent Semantic Indexing (LSI) [6] or the Latent Dirichlet Allocation (LDA) [4]. Our use of the n-gram graphs causes a transformation into a space defined by graph similarities, and not latent topics. These graph similarities correspond to a representation of a class through the class graph.

To face sparseness, content-based representation models are typically combined with the context-based ones, which incorporate data collected outside the training set. In [28], an external "universal dataset" is employed to help determine a set of hidden topics using LDA. The empirical evaluation verifies a significant increase in classification performance, provided that a good universal dataset as well as good LDA parameters (e.g., number of hidden topics) have been selected. In [38], the authors propose the application of Transductive (i.e., test-set-tailored) LSI to classify short texts, demonstrating that evidence extracted from the test instances can improve the performance of classification. In [34], Wikipedia-based "explicit semantic analysis" is used to map sparse text snippets (e.g., from ads or "tweets") to Wikipedia concepts. In another line of research, [18] studies the effect of different topic modeling methods on the classification of tweets in the presence of data sparseness; it advocates that aggregating short messages and performing topic modeling on them improves performance. Another effective approach is to use author-related information to augment the set of features of tweets [32]. In our work we use no external source of knowledge to augment the data. Thus, such approaches are orthogonal to ours and can be combined with the n-gram graph framework to further improve its performance.

Another aspect of text classification relates to Social Media content, which involves many intricate characteristics for TC: multi-lingual content, very short and sparse texts, fully evolving and non-standard vocabulary, noise as well as lack of labeled resources [24]. In [25], the authors apply text classification as a keyword extraction process from "social snippets" (e.g., status updates, interesting events or recent news), using a variety of features like TF-IDF, and linguistic, position and formatting information. In [19], a system is proposed to detect spam tweets in what the authors call the "trend stuffing". The classification is essentially binary, determining whether a tweet is related to a highly active topic (i.e., "trend") or not. In [14], the authors use external knowledge, mapping each tweet to Wikipedia to define a measure of semantic relatedness between pairs of tweets based on the links between Wikipedia pages. [23] considers an alternative source of external knowledge: the metadata of linked objects appearing in Social Media posts. These metadata are used to augment the feature space of the posts or even completely replace them. The results of the experimental study indicate that they can improve topic classification of posts, even if they are used without the original content-based features. In our work, we use corpora from different document types to illustrate their differences. We use only content features, with no linguistic preprocessing (graph creation) and no external knowledge.

7. CONCLUSIONS

In this work, we presented novel points of research in topic classification on several axes. We provided a set of differentiating criteria between textual types, and offered insight on the aspects of Web documents that affect the performance of text classifications. We took into account a multi-lingual setting, which is important for general application. We proposed the use of a non-standard representation (i.e., n-gram graphs) that — in comparison with traditional document models — conveys not only higher effectiveness (i.e., classification accuracy), but also higher efficiency in the learning process. This is achieved through a limited set of expressive features, whose cardinality actually depends on the number of classes, rather than on the diversity of the vocabulary. Our experimental study comprised three large-scale, real-world corpora — one per document type — thus stressing the different challenges among the main types of Web documents. We did not use external knowledge and we minimized language-dependency. We demonstrated that the proposed method can provide better results than the traditional ones in these challenging settings, due to the contextual information it encapsulates.

In the future, we intend to examine how the adaptive nature of n-gram graphs can accommodate the evolution of the discriminative features of a topic with the passage of time (i.e., *topic drift*), a phenomenon that is particularly intense in the real-time content of Social Media. In addition, we plan to compare the n-gram graphs model with the state-of-the-art context-based ones as well as with approaches based on LDA, possibly aiming to combine the power of n-gram graphs with a generative model.

Acknowledgement

This work has been supported by the SocIoS project and has been partly funded by the European Commission's 7th Framework Programme through theme ICT-2009.1.2: Internet of Services, Software and Virtualisation under contract no.257774.

References

[1] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. Finding high-quality content in social media. In *WSDM*, pages 183–194, 2008.

[2] M.-R. Amini, N. Usunier, and C. Goutte. Learning from multiple partially observed views - an application to multilingual text categorization. In *NIPS*, pages 28–36, 2009.

[3] M. W. Berry and J. Kogan. *Text Mining: Applications and Theory*. Wiley, 2010.

[4] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[5] A. Dasgupta, P. Drineas, B. Harb, V. Josifovski, and M. W. Mahoney. Feature selection methods for text classification. In *KDD*, pages 230–239, 2007.

[6] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[7] S. Dumais and H. Chen. Hierarchical classification of web content. In *SIGIR*, pages 256–263, 2000.

[8] J. Eisenstein, B. O'Connor, N. A. Smith, and E. P. Xing. A latent variable model for geographic lexical variation. In *EMNLP*, pages 1277–1287, 2010.

[9] H. Escalante, T. Solorio, and M. Montes-y Gómez. Local histograms of character n-grams for authorship attribution. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 288–298. Association for Computational Linguistics, 2011.

[10] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[11] F. Figueiredo, F. Belém, H. Pinto, J. M. Almeida, M. A. Gonçalves, D. Fernandes, E. S. de Moura, and M. Cristo. Evidence of quality of textual features on the web 2.0. In *CIKM*, pages 909–918, 2009.

[12] G. Forman. An extensive empirical study of feature selection metrics for text classification. *The Journal of Machine Learning Research*, 3:1289–1305, 2003.

[13] S. Garcia Esparza, M. O'Mahony, and B. Smyth. Towards tagging and categorization for micro-blogs. In *AICS*, 2010.

[14] Y. Genc, Y. Sakamoto, and J. V. Nickerson. *Discovering Context: Classifying Tweets through a Semantic Transform Based on Wikipedia*, pages 484–492. 2011.

[15] G. Giannakopoulos, V. Karkaletsis, G. A. Vouros, and P. Stamatopoulos. Summarization system evaluation revisited: N-gram graphs. *TSLP*, 5(3), 2008.

[16] G. Giannakopoulos and T. Palpanas. Content and type as orthogonal modeling features: a study on user interest awareness in entity subscription services. *International Journal of Advances on Networks and Services*, 3(2), 2010.

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[18] L. Hong and B. Davison. Empirical study of topic modeling in twitter. In *SOMA*, pages 80–88, 2010.

[19] D. Irani, S. Webb, C. Pu, and K. Li. Study of trend-stuffing on twitter through text classification. In *CEAS*, 2010.

[20] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. pages 137–142, 1998.

[21] I. Kanaris, K. Kanaris, I. Houvardas, and E. Stamatatos. Words versus character n-grams for anti-spam filtering. *International Journal on Artificial Intelligence Tools*, 16(6):1047, 2007.

[22] A. Khorsi. An overview of content-based spam filtering techniques. *Informatica*, 31:269–277, 2007.

[23] S. Kinsella, A. Passant, and J. G. Breslin. *Topic Classification in Social Media Using Metadata from Hyperlinked Objects*, pages 201–206. 2011.

[24] S. Kinsella, M. Wang, J. G. Breslin, and C. Hayes. Improving categorisation in social media using hyperlinks to structured data sources. In *ESWC (2)*, pages 390–404, 2011.

[25] Z. Li, D. Zhou, Y. F. Juan, and J. Han. Keyword extraction for social snippets. In *WWW*, pages 1143–1144, 2010.

[26] C. Manning, P. Raghavan, and H. Schuetze. *Introduction to information retrieval*, volume 1. Cambridge University Press, 2008.

[27] A. Pak and P. Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. *Proceedings of LREC 2010*, 2010.

[28] X. H. Phan, M. L. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *WWW*, pages 91–100, 2008.

[29] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc. New York, NY, USA, 1986.

[30] F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, 2002.

[31] F. Sebastiani. Text categorization. In *Encyclopedia of Database Technologies and Applications*, pages 683–687. 2005.

[32] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. Short text classification in twitter to improve information filtering. In *SIGIR*, pages 841–842, 2010.

[33] E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Automatic text categorization in terms of genre and author. *Computational linguistics*, 26(4):471–495, 2000.

[34] X. Sun, H. Wang, and Y. Yu. Towards effective short text deep classification. In *SIGIR*, pages 1143–1144, 2011.

[35] T. Wilson and S. Raaijmakers. Comparing word, character, and phoneme n-grams for subjective utterance recognition. In *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[36] I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[37] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *WSDM*, pages 177–186, 2011.

[38] S. Zelikovitz and H. Hirsh. Transductive lsi for short text classification problems. In *FLAIRS*, pages 556–561, 2004.