The Knowledge Engineering Review, Vol. 00, e0, 1 of 38. © Cambridge University Press, 2019
doi:10.1017/S0269888919000018

Short-text learning in social media: a review

Antonela Tommasel and Daniela Godoy

ISISTAN, UNICEN-CONICET, Campus Universitario, Tandil (B7001BBO), Argentina;
e-mail: antonela.tommasel@isistan.unicen.edu.ar, daniela.godoy@isistan.unicen.edu.ar

Abstract

Social networks occupy a ubiquitous and pervasive place in the life of their users. The substantial amount of content generated and shared by social networking users offers new research opportunities across a wide variety of disciplines, including media and communication studies, linguistics, sociology, psychology, information and computer sciences, or education. This situation, in combination with the continuous growth of social media data, creates an imperative need for content organisation. Thus, large-scale text learning tasks in social environments arise as one of the most relevant problems in machine learning and data mining. Interestingly, social media data pose several challenges due to its sparse, high-dimensional and large-volume characteristics. This survey reviews the field of social media data learning, focusing on classification and clustering techniques, as they are two of the most frequent learning tasks. It reviews not only new techniques that have been developed to tackle the new challenges posed by short-texts, but also how traditional techniques can be adapted to overcome such challenges. Then, open issues and research opportunities for social media data learning are discussed.

1 Introduction

Social networking sites, such as *MySpace*, *Facebook* or *Twitter* occupy a ubiquitous and pervasive place in the life of their millions of users. As a result, social media data grow at an unprecedented rate. The substantial amount of content generated and shared by social networking users offers new research opportunities across a wide variety of disciplines, including media and communication studies, linguistics, sociology, psychology, information and computer sciences, and education. For example, social media can help to solve communication and coordination problems that might arise due to geographical distances in case of extreme events or emergencies (Li *et al.*, 2011b), or they can increase the effectiveness of social campaigns by helping with the dissemination of the required information (Li *et al.*, 2011a). This situation, in combination with the continuous growth of social media data, creates the imperative need of organising the content. As a result, large-scale text learning tasks in social environments arise as one of the most relevant problems in machine learning and data mining.

Text mining refers to a knowledge discovery process aiming at the extraction of interesting and non-trivial patterns from natural language. This process includes multiple fields, such as text analysis, natural language processing and information retrieval, amongst others. Text mining and learning tasks are characterised by the high dimensionality of their feature space where most terms have a low frequency. Indeed, text learning is often susceptible to the problem known as the 'curse of dimensionality', which refers to the increasing computational complexity of learning tasks as the data that need to be accessed grow exponentially regarding the underlying space dimension. Furthermore, as data dimensionality increases, the volume of the feature space grows rapidly, so that the available data become sparse.

With the advent of short-texts, new challenges in automatic learning processes are faced. First, unlike traditional and long texts, in the context of social media, text are usually noisier, less topic-focused and shorter (e.g., tweets can be up to 280 characters long). Moreover, users generally communicate through unstructured or semi-structured language. Users neglect the usage of accurate spelling or grammatical structures, which might lead to different types of lexical, syntactic or semantic ambiguities. Traditional text learning approaches compute the similarity between two documents (e.g., posts, tweets or Web snippets) heavily relying on term overlapping. Due to the low frequency of terms appearing in short-texts, similarity metrics might not be suitable for assessing short-text resemblance (Ni *et al.*, 2011). Thus, traditional approaches solely based on text resemblance might not achieve high performance when applied to short-texts, which increases the complexity of learning tasks. Second, the linked nature of social media data generates new information, such as who is sharing the posts (authorship, i.e., user-post relations), and who is friend of whom (friendship, i.e., user-user relations), which can be added to the feature space (Tang & Liu, 2012). Third, the continuous growth of social media data puts in jeopardy the scalability of current techniques (Tang *et al.*, 2014), especially in real-time tasks. For example, most techniques require all data to be loaded into memory, which might not be possible when analyzing high volumes of social media data. In response to these challenges, new learning approaches specifically designed for short-texts have emerged.

This survey reviews the field of short-text learning, focusing on classification and clustering techniques, as they are two of the most frequent learning tasks. Other surveys in the literature (Song *et al.*, 2014; Irfan *et al.*, 2015) focus either on presenting a taxonomy of techniques attempting at providing an understanding of the textual patterns in social media (Irfan *et al.*, 2015), or on describing semantic feature reduction techniques as well as describing a few techniques and characterising the evaluation metrics and processes (Song *et al.*, 2014), thus ignoring the challenges and problems of social media learning. Instead, this survey illustrates how the classification and clustering of short-texts (with a special emphasis on social media data) have been tackled, discussing limitations and current unsolved issues of the existing techniques. In this regard, a comprehensive view of state-of-the-art approaches for short-text learning is given, identifying their advantages and limitations.

The rest of this survey is organised as follows. Section 2 presents general concepts related to text learning techniques. Sections 3 and 4 review classification and clustering techniques, respectively. The reviewed techniques are classified based on whether they are traditional learning techniques that have been applied to short-text data, they represent adaptations or modifications of traditional learning techniques made for coping with the particularities of short-texts, or they are specifically designed for dealing with texts from social media sites. Although the adaptations of traditional learning techniques could also be regarded as new techniques, the differentiation aims at distinguishing between simple modifications that do not change the nature of techniques, and techniques that present more profound modifications implying a specific combination of techniques, add new steps to traditional techniques or introduce new learning approaches that are independent from already known ones. Then, the new techniques are organised based on how they tackle the problem of short-text learning. Considering the reviewed works, Section 5 analyzes open issues and research opportunities for performing learning over social media texts. Finally, Section 6 presents the conclusions of the survey.

2 An overview of short-text learning

The automated classification of texts into pre-defined categories dates back to 1960. In the last decade, it experimented a growing interest due to the increasing document availability in digital forms and the imperative need to organise them (Sebastiani, 2002). Text represents a specific kind of data in which word attributes are sparse and highly dimensional, and frequencies are low for most words (Aggarwal & Zhai, 2012). Almost all known learning techniques have been effectively used on text data (Sebastiani, 2002). However, with the advent of short-texts and social media data, new challenges in the automatic learning process need to be faced, especially in resource-constrained scenarios (Aggarwal, 2014).

Short-texts hinder learning tasks due not only to their sparseness and the reduced word frequencies, but also to the fact that their topics change constantly at a fast rate, thus requiring new training data. Moreover, traditional techniques might not achieve high performance when applied to short-texts. Consequently, short-text learning tasks are regarded as more complex than their long-text counterpart.

Additional challenges are posed by the high volume of available data, which continuously appears in the form of streams (Gandomi & Haider, 2015). Thereby, a growing need for real-time analysis arises. For example, the real-time processing of tweets can be applied to the detection of traffic incidents, natural disasters or to the prevention of the diffusion of misinformation and disinformation in the occurrence of an event. The problem of text stream learning can arise in two different scenarios (Aggarwal, 2014), depending on whether there are sufficient instances for learning a model. In the first case, training instances might be available for batch learning, but new instances might arrive in the form of a stream. In the second case, both training and test instances are not known in advance, as they arrive in the stream. The first scenario could be easy to handle as once a model is learned with the known instances, most learning approaches can classify or cluster individual instances efficiently. Conversely, in the second scenario, learned models need to be incrementally updated to account for changes in the constantly arriving training data. Moreover, in these types of scenarios it might be impractical, time-consuming or impossible to label the entire stream for training. This situation fosters the appearance of the phenomenon known as concept drift (Lifna & Vijayalakshmi, 2015), by which concepts appear and disappear often, affecting the temporal relevance of data. Nonetheless, although learning on data stream scenarios is crucial in real-life applications, it has received little attention.

Text learning tasks are usually accompanied by feature selection techniques, which represent other crucial problem in machine learning. Feature selection (Alelyani *et al.*, 2013) is one of the most known and commonly used techniques for reducing the high-dimensional feature space by removing redundant and irrelevant features. Dimensionality reduction helps both to speed up data mining algorithms and to improve mining performance (Liu & Yu, 2005). A wide variety of feature selection methods have been proposed in the literature. Feature selection algorithms can be regarded as a combination of a search technique for finding feature subsets with an evaluation measure that scores the different feature subsets (Guyon & Elisseeff, 2003). These algorithms are traditionally organised into four categories depending on how the feature subset is selected (Liu & Yu, 2005; Saeys *et al.*, 2007; Alelyani *et al.*, 2013): filter, wrapper, hybrid and embedded. Filter techniques consider the intrinsic statistical characteristics of features independently of any classifier. Wrapper techniques select the feature subset with the highest discriminative power regarding a specific learning algorithm, which makes them more computationally complex than filter techniques. Hybrid techniques first use statistical criteria to select candidate features subsets with a specific cardinality, and then choose the subset with the highest performance according to a learning algorithm. Finally, embedded techniques perform feature selection simultaneously to other data mining tasks. As the search for the best feature subset is built into the construction of a learning model, embedded methods are also specific to a given algorithm.

Feature selection techniques can be used in combination with both supervised and unsupervised learning models. Unsupervised feature selection is performed when the class labels of instances are unknown. In contrast, supervised feature selection techniques select a subset of highly discriminant and relevant features guided by class information. Such feature subset should be useful for discriminating instances belonging to different classes. However, obtaining such labelled instances could be time consuming. The problem worsens in online environments in which short-texts are constantly generated. Whilst unsupervised feature selection considers unlabelled data, it could be difficult to accurately assess the relevance of features. Particularly, when considering social media texts, it is common to have an enormous volume of high-dimensional data, but a small volume of labelled instances. Consequently, feature selection techniques have to be carefully designed to cope with these characteristics.

2.1 Short-text characterisation

Short-texts can adopt multiple formats, such as mobile short messages, chat messages, news titles, blog comments, social media comments, tweets and *Facebook* posts, amongst other possibilities. The characteristics of short-texts can be summarised as follows (Weller *et al.*, 2013):

- *Sparseness*. Short-texts only contain a few terms in the context of a continuously expanding vocabulary. Hence, they might not provide enough terms to accurately compute term co-occurrence or similarity.

https://doi.org/10.1017/S0269888919000018

- *Immediacy*. A great volume of short-texts are continuously created, generally appearing in the form of real-time streams.
- *Non-standardability*. Unlike traditional and long-texts, in the context of social media, text are usually noisier. Users generally communicate through unstructured or semi-structured language. Moreover, users neglect the usage of accurate spelling and grammatical structures, which might lead to different types of lexical, syntactical or semantic ambiguities. Hence, it might be difficult to extract valid language features.
- *Large scale—scalability*. The continuous growth of social media data puts in jeopardy the scalability of current learning techniques (Tang *et al.*, 2014), especially in real-time tasks.
- *Limited-labelled examples*. When considering social media texts, it is common to have an enormous volume of data, but a small volume of labelled instances. In this regard, techniques adequately dealing with an unbalanced set of training instances are needed.
- *Persistence*. Online short-texts in social media are automatically recorded and archived (with a few exceptions, such as *Snapchat*[1]).
- *Availability–searchability*. Social media content is globally available. Almost every short-text available in social public networks can be accessed through search.

2.2 Applications of short-text learning

Text learning has applications in a wide variety of domains (Sebastiani, 2002; Aggarwal & Zhai, 2012):

- *News filtering and organisation*. Social media sites have emerged as a fast communication channel for spreading news, predicting political results and sharing political events and conversations. As a result, a large number of news posts is created every day, hindering their manual organisation (Aggarwal & Zhai, 2012).
- *Event detection*. Social media posts often reflect the existence of events as they happen. Event detection in real-time is a challenging problem due to the heterogeneity and immense scale of data (Becker *et al.*, 2011). Events can range from large social events (e.g., music concerts) to natural disasters. Moreover, social media has also been used as an important tool for crime prediction and terrorist monitoring (Oh *et al.*, 2011).
- *Opinion mining*. Companies are increasingly using social media sites to advertise and recommend their services and products, build their reputation, respond to customers' complaints or to improve decision making (Khan *et al.*, 2014). In this context, customer reviews or opinions are often short-text documents, which can be mined to determine useful information from the review.
- *Sentiment analysis*. It has been demonstrated that public sentiments extracted from social media can be used to predict real-world outcomes (Asur & Huberman, 2010). Additionally, correlations have also been found between the raise of new topics and changes in sentiment (Thelwall *et al.*, 2011).
- *Spam or rumour filtering*. Whilst the spread of inaccurate information has always been a concern, the ubiquitous usage of social media has exacerbated the problem by facilitating the spread of such information on large user communities (Zubiaga *et al.*, 2015; Ciampaglia *et al.*, 2015). This has special consequences in the case of emergencies, in which the spread of false rumours can have dangerous consequences.

3 Classification

Classification is the problem of identifying to which class a new data instance belongs to, on the basis of a training dataset containing instances whose class membership is already known (Tang *et al.*, 2014). Figure 1 depicts the general classification methodology for a batch scenario, which can be defined as follows (Aggarwal & Zhai, 2012). There exists a set of training instances comprising features. For example, in the context of social media, instances represent short-texts and the terms in each of them their features.

[1] https://www.snapchat.com/l/es/.

https://doi.org/10.1017/S0269888919000018

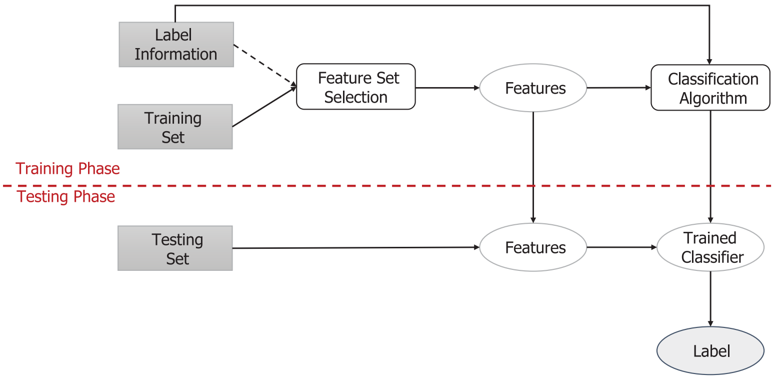

Figure 1 Overview of the batch classification process

Additionally, each instance is labelled with a class value drawn from a discrete set. Features can be either categorical, ordinal, integer-valued, real-valued or strings. Nonetheless, not every classification algorithm leverages on all feature types. A feature selection process might be applied to restrict the set of features to be used. Note that the label information of instances might be needed for the selection of features. Then, all training data are used to build a classification model, which associates the features in the instances with the class labels. As all training data are used at once, problems related to the large-scale processing of data could appear. Finally, the class of test instances (i.e., instances whose class is unknown) can be predicted for evaluation. Test instances are first represented by the feature set extracted in the training process, and then the trained model is used to predict a class label. There are several possibilities for assigning a class to an instance. New instances are either explicitly assigned to a single (or multiple) class, or a probability of belonging to the different classes is computed.

In the case of an incremental scenario (Losing *et al.*, 2018), the learning process is modified to accept one training instance at a time. Thus, the learned model is updated as each new instance is known in a pre-defined order. The main challenge is not the large-scale processing of data, but the continuous and efficient learning from few data instances. Finally, once all training instances are known, the last trained model is used to predict the class of the test instances. This scenario only allows to evaluate the generalisation ability of the last trained model, neglecting all the preceding models. Thus, it is useful in those cases in which a lot a training data are available to continuously construct a model as accurate as possible. Finally, social media applications might involve an online scenario (Losing *et al.*, 2018), in which instances are not split into training and test scenario. The main difference with the incremental scenario is that all intermediate models are considered for the performance evaluation, but each intermediate model only predicts the following instance. In this regard, each instance is initially used for model testing and then for updating the learned model.

3.1 Traditional classification techniques

Almost all known classification techniques, such as decision trees, decision rules, Bayesian methods, k-nearest neighbours (k-NN), support vector machines (SVM) and neural networks, amongst others, have been used to effectively classify text data (Sebastiani, 2002). Rosa and Ellen (2009) experimentally evaluated traditional text classification techniques in the context of military short-texts. Four different classification algorithms were evaluated: SVM, k-NN, Rocchio and naïve Bayes (NB). The selected dataset comprised a medium-scale number of chat lines extracted from a US military chat. Each line contained a message in the form of text, timestamp and a randomly selected author. Additionally, noisy posts were randomly added to the dataset, and randomly selected filler posts were added a random naval ship key to simulate posts that are not tactically significant, but contain terms that appear in posts belonging to other categories. When including all features, k-NN obtained the best overall results. Nonetheless, as the number of selected features decreased, SVM and NB improved their performance. The authors concluded that k-NN and SVM are useful for categorising informal short-texts.

https://doi.org/10.1017/S0269888919000018

Rosa *et al.* (2011) analyzed the performance of the Rocchio classifier when applied to short-texts, such as tweets. Tweets were classified into general or specific categories according to their hashtags. Experimental evaluation was based on a medium-scale number of tweets, which were pre-processed by lower-casing terms, removing terms appearing less than five times, removing all non-alphanumeric characters and mapping URLs and usernames to two special terms. Classification into the generic categories achieved better performance than classification into the specific ones. Considering the different pre-processing strategies applied, the authors found that removing all words occurring less than five times did not significantly improve classification performance when compared to only lower-casing terms. However, it helped to significantly reduce the dimensionality of the feature space, which, in turn, reduced the computational complexity of the classification. On the contrary, removing all non-alphanumeric characters and replacing URL and usernames degraded the performance, as it led to the loss of valuable contextual information. Finally, the authors compared classification performance with the clustering performance of k-Means and latent Dirichet allocation (LDA). Although k-Means outperformed LDA results, both algorithms produced poor quality clusters regarding their topic distribution as they tended to cluster tweets based on language similarity rather than topical similarity. In this context, classification proved to be more useful than clustering to identify tweet's topics.

Regarding spam classification in mobile texts messages, both Yu and Chen (2012) and Mathew and Issac (2011) analyzed the performance of traditional text classification techniques. Yu and Chen (2012) proposed a binary NB classification for online spam short message filtering based on the content of messages and social network features. Traditional feature selection techniques such as information gain (IG) and odds ratio (OR) were used. Results showed that the performance achieved when considering *IG* was not affected by variations in the number of selected features, whereas the performance of OR was affected by them. Additionally, combining IG and OR achived better results than the individual techniques. Finally, adding social features also improved results of considering only textual features. The authors concluded that short-texts can be effectively classified by means of NB. Nonetheless, the authors did not compare the obtained results with any baseline nor state-of-the-art classifier.

Mathew and Issac (2011) proposed an offline spam classification system. Experimental evaluation was based on 32 algorithms selected from Weka,[2] such as NB, Bayes net and sequential minimal optimisation (SMO), amongst others. Results showed that Bayesian methods achieved the best performance in most cases, obtaining up to 98% of correct classifications and few false positives. SMO was also amongst the best performing algorithms. However, it was also one of the most time consuming algorithms. The authors detected seven algorithms (multilayer perceptron, radial basis function—RBF—network, simple logistic and lazy Locally Weighted Learning (LWL), amongst the most common ones) that showed not to be useful due to excessive memory consumption or long execution times. The authors concluded that Bayesian methods were one of the best approaches for mobile messages classification, confirming the findings in (Yu & Chen, 2012). In summary, Rosa and Ellen (2009) concluded k-NN and SVM are useful for classifying informal short-texts, whereas Mathew and Issac (2011) stated that Bayesian methods and SMO are amongst the best performing algorithms for mobile message classification.

Ensemble classifiers are learning techniques that train a set of classifiers and then classify the new instances by considering a weighted vote of their predictions (Dietterich, 2000), aiming at reducing the bias and noise of individual classifiers. In this context, Prusa *et al.* (2015) evaluated the performance of two ensemble techniques (bagging and boosting) in combination with seven classification algorithms (two versions of the C4.5 decision tree classifier, multilayer perceptron, k-NN considering five neighbours, SVM, RBF network and logistic regression) for sentiment classification in *Twitter*. Results showed that considering an ensemble technique significantly improved the performance of most learners, with the exception of k-NN. Although the results obtained with bagging were higher than those obtained with boosting, differences were not statistically significant. The highest results were obtained for RBF, followed by SVM. Interestingly, considering SVM alone achieved equal or better performance than most of the other learners when applying an ensemble technique, confirming the superiority of SVM for classifying short and informal text.

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

Given that in social media it might be difficult to collect a balanced number of labelled instances, Prusa *et al.* (2016) evaluated ensemble learners on class imbalanced scenarios. The authors analyzed the effect of performing data sampling on ensemble techniques for tweet sentiment analysis. Four base classifiers were selected (C4.5 decision tree, NB, SVM and logistic regression) and combined with boosting ensemble. Based on the results in (Li *et al.*, 2011c), data sampling was performed by means of random undersampling as it was shown to perform better than oversampling. Feature selection was performed considering the area under the ROC curve and selecting several subsets of different sizes. Evaluation was based on the sentiment140 *Twitter* corpus,[3] from which two samples with moderate (20:80 class ratio) and severe (5:95 class ratio) levels of class imbalance were selected. Results showed that including data sampling allowed to significantly improve the ensemble classifier results for most combinations of learners and two feature subset combinations. The biggest differences were observed for the most imbalanced dataset. Regarding the base classifiers, the biggest improvements were observed for C4.5, whilst the smallest for SVM. The authors concluded that training a robust classifier (such as SVM) using boosting might be sufficient for datasets with low levels of class imbalance, nonetheless data sampling is needed for higher imbalance levels.

Unlike the previous works that focused on single label classification, Sajnani *et al.* (2011) evaluated multi-label classification techniques in the context of short-texts. The study focused on four classification algorithms (k-NN, NB, SVM and multilayer perceptron) in combination with two types of transformations to multi-label scenarios (binary relevance and label powerset). Results were compared to those of adapting the traditional algorithms to handle multi-label data directly. Three feature sets were defined: all textual features, features with a frequency higher than 6 and contextual features. Interestingly, the best results were observed for k-NN and NB, regardless of the considered multi-label transformation, whilst the worst results were observed for SVM. Overall, the best results were obtained when considering binary relevance, followed by the multi-label adaptation of algorithms. Nonetheless, the authors did not perform any analysis to verify the statistical superiority of the technique.

3.2 Enhancement to traditional classifiers

Several works (Yuan *et al.*, 2012; Kim *et al.*, 2014) focused on enhancing traditional classifiers to improve their performance over short-texts by modifying the underlying probabilistic models (Yuan *et al.*, 2012) or designing new similarity functions (Kim *et al.*, 2014). Yuan *et al.* (2012) analyzed the performance improvements of NB in short-text classification by considering four smoothing techniques that modify how class probabilities are computed. In particular, the authors applied the Jelinek–Mercer, Dirichlet, absolute discounting and two-stage smoothing techniques (Zhai & Lafferty, 2004). Evaluation was performed on a question topic classification task, including approximately 4 million questions belonging to 1097 classes. Results showed that applying smoothing to NB could improve its performance, achieving similar results to SVM, but with a simpler and more efficient approach.

Kim *et al.* (2014) proposed a language independent semantic (LIS) similarity function based on both syntactic and semantic text features that was also independent of any grammatical tag and lexical databases. LIS comprises three steps: pattern extraction, semantic annotation and similarity computation. Syntactic patterns are defined as sets of words appearing in texts based on the syntactic information of a specific language. Particularly, the authors chose to extract patterns based on syntactic parse trees as they provide information about both the occurrence and sequence of words. Then, semantic information is annotated on each extracted pattern by considering three semantic levels: word (words that frequently co-occur within a pattern are used to define its meaning in word-level annotations), document (the meaning of each pattern is expanded with the words appearing in similar documents in the document-level annotation) and class (patterns are associated with a class based on their similarity). Experimental evaluation was based on English and Korean short-texts, out of which the majority contained at most 20 words and approximately 23% of them contained less than 5 words. LIS was added to the SVM algorithm and compared to the traditional Bag Of Words (BOW) model (weighted with term frequency—TF and TF

[3] http://help.sentiment140.com/home.

https://doi.org/10.1017/S0269888919000018

Inverted Document Frequency—TF-IDF), string, and syntactic semantic tree similarity metrics. Results showed that LIS outperformed the accuracy of the other three similarity metrics regardless of the number of classes involved. However, when short-texts comprised less than 5 words, String similarity obtained the best results. Additionally, BOW outperformed LIS when considering texts with more than 25 words.

Finally, Su-zhi and Pei-feng (2011), Cui *et al.* (2016) and Collins *et al.* (2015) proposed alternatives to combine the outputs of classifiers. Su-zhi and Pei-feng (2011) presented a hybrid classification technique (named SVM-KNN or KSVM for short) designed to adopt either SVM or k-NN according to specific distributions of samples in the space. If the distance of a new instance to the separating hyperplane is greater than a pre-defined threshold, SVM is assumed to accurately classify the instance. Otherwise, SVM would only compute the distance to the one representative point (i.e., support vectors), which could lead to inaccurate classifications. In those cases, k-NN measures the distance to all the representative points by means of the Cosine distance. Experimental evaluation was based on 10 567 comments extracted from *TianYa BBS*,[4] *Sohu*[5] and *Sina News*[6] from five categories. The new algorithm was compared to SVM and k-NN by means of recall, precision and F-measure. Results showed that the recall and precision of KSVM were above 84%, outperforming both SVM and k-NN by approximately 2%. Additionally, both SVM and k-NN obtained similar results with variations lower than 0.5%.

Cui *et al.* (2016) used an additional learning layer for combining the outputs instead of a weighted combination of probabilities for tweet topic classification. They included three classifiers in the ensemble: labelled-latent Dirichlet allocation, NB and a new classifier based on a cloud taxonomy service. Instead of basing the ensemble approach on traditional probabilistic Bayesian voting models, the authors introduced a stacking classifier and ensemble approach. Such classifier combines the output from the individual models to generate the final decision. Unlike other ensemble methods that treat the probabilities of the individual models as indicators of reliability, the stacking method considers them as a representation of the instances provided by the individual models, which is then used to train the classifier. The probabilistic representation of instances is combined with an n-gram vector of tweets. The authors choose the maximum entropy classifier as the stacking classifier. Results were compared with widely used probability-based ensemble methods such as Weighted Sum and Product of Experts. According to the authors, their approach achieved the best performance, whilst showing to better adapt to the variations of the probability distributions from the individual classifiers. Nonetheless, the variations of accuracy were lower than 2%. Given the small differences, an analysis of computational complexity is also needed.

Collins *et al.* (2015) proposed a sequence of three classifiers for performing sentiment analysis. The initial classifier is an SVM, which classifies tweets as positive, negative or neutral. Then, a rule-based classifier is introduced to reduce the number of tweets that are incorrectly labelled as negative. This classifier counts the number of positive and negative words in the tweet according to a sentiment lexicon (Hu & Liu, 2004). For those tweets labelled as negative, if the number of positive words is greater than that of negative words, the label is changed to neutral. On the other hand, for those tweets labelled as positive or neutral, two NB classifiers are used. The first is trained only on positive and neutral tweets, whilst the second is trained on all tweets and labelled them as either neutral or non-neutral. In those cases where a non-neutral label is predicted, the rule-based classifier is used to determine the final labelling. Evaluation was performed in the context of the SemEval-2015[7] competition, achieving competitive results.

3.3 New classification techniques

Several works (Nishida *et al.*, 2011; Romero *et al.*, 2013; Ramírez de la Rosa *et al.*, 2013; Yang *et al.*, 2013; Zhang & Zhong, 2016; Wang *et al.*, 2016a; Dai *et al.*, 2017; Li *et al.*, 2018; Ravi & Kozareva, 2018; Yan *et al.*, 2018) focused on designing new classification techniques specifically for

[4] http://bbs.tianya.cn/.

[5] http://www.sohu.com/.

[6] http://news.sina.com.cn/.

[7] http://alt.qcri.org/semeval2015/.

https://doi.org/10.1017/S0269888919000018

short-texts. The reviewed techniques are organised under three categories, according to how they tackle the classification task: word/character sequence-based techniques (including techniques based on data compression and similarity), domain knowledge-based techniques (including ontologies and diverse corpora) and neural networks-based techniques (including techniques leveraging on word embeddings and deep learning).

3.3.1 Word/character-based techniques

Several authors aimed at classifying short-texts by proposing new techniques that leverage on the characteristics of the languages in which texts are written (Nishida *et al.*, 2011), or on adapting the concept of similarity to deal with the brevity, sparseness or temporal relevance of short-texts (Ramírez de la Rosa *et al.*, 2013; Sedhai & Sun, 2018).

Considering the differences between languages regarding the importance of word order in a sentence or the chosen delimiter character (e.g., Japanese do not use whitespace for delimiting words, whilst English does), Nishida *et al.* (2011) proposed a technique based on data compression for binary short-text classification. The technique is language independent and capable of effectively leveraging on the context of terms. In the first step, it analyzes the compressibility of texts regarding the positive and negative examples, and then computes the classification score. Compressibility was evaluated by the standard Deflate algorithm (Deutsch, 1996), which is used by gzip. As Deflate is not a character-based algorithm, texts from the same language might have similar compression rates. Therefore, the last byte of each non-ASCII character is considered to reduce the compression variance rate of different languages. Although the authors selected Deflate, any data compression algorithm could be used. However, the generalisation of results cannot be guaranteed as they might depend on the intrinsic characteristics of each algorithm. The models learned for the positive and negative classes are updated by concatenating the most recent instances. This implies that the technique could be used in online or streaming environments in which instances continuously arrive. Then, a text is assigned to the positive class if its compressibility value is lower than a pre-defined threshold. Lower thresholds improve the precision of classifications, whereas higher thresholds improve their recall. The compressibility value is defined as the ratio between the indicators of text comprehensibility of the positive and negative classes. Experimental evaluation was based on more than a million tweets of multiple languages containing a single hashtag out of the four selected by the authors. Results showed that the proposed technique significantly outperformed other algorithms in terms of precision and recall. Nonetheless, comparisons might have not been fair, as the selected gram representation for baselines might not be adequate in the case of Japanese texts. Moreover, the quality of classifications varied according to the selected hashtag. For example, the worst results were observed for the most polysemous one, which might imply the sensitiveness of the approach to new senses or changes in the structure of texts. This situation might hinder the applicability of the technique in social media environments, in which short-texts are sparse and topics can constantly appear and disappear.

As mentioned, short-texts do not only hinder classification tasks due to their sparseness and the reduced word frequencies, but also because their topics change constantly at a fast rate, thus regularly requiring new training data. In this context, Ramírez de la Rosa *et al.* (2013) proposed a neighbourhood consensus classification technique that is based on the idea that similar documents might belong to the same category. Texts are classified by considering their own information and that of the category assigned to other similar texts in the same test collection. This technique differs from others in that it does not modify the training set nor employs a target-collection of texts to build the classification model. Instead, such information is used to support the classification decision made by a weak classifier. The technique consists of two steps. The first, known as training, carries out the construction of the classifier using a set of labelled documents. Any supervised classification algorithm can be used for this step. The second, known as classification, identifies the most similar texts (i.e., the neighbours) to each text from the target collection, and then assigns it to a class. The authors chose to apply a modified version of the prototype-based classifier. In the training step, a single representative instance (i.e., the prototype) for each class is built. In the classification step, the similarity scores for each unlabelled document and class prototype are computed. Then, a combined similarity score for each unlabelled text is computed as the linear combination of the similarity score between the text and the class prototype, and the similarity between the texts with the same class prototype. Finally, the text is assigned to the class with the largest combined similarity score. The contribution of each neighbour to the linear combination of similarities is inversely proportional to the text of interest (as in the k-NN) with the main difference that neighbours are the other unlabelled texts and not the texts in the training set. Experimental evaluation was based on both short- and long-texts. The short-texts comprised the titles of news articles with less than eight words. The performance of the approach was compared to that of SVM, k-NN, C4.5 and NB in terms of macro F-measure. Results showed that the proposed approach significantly outperformed the other classifiers when considering short-texts. The greatest differences were obtained regarding SVM and NB. Additionally, the smaller the training set, the greater the improvements, which could indicate that the proposed approach might be useful in situations in which labelled examples are scarce. On the contrary, the approach did not significantly improve the other classifiers when considering long-texts. Moreover, the accuracy of the approach remains to be evaluated in the context of social media. As stated by the authors, any classification algorithm could be selected for the first step of the technique, implying that an incremental algorithm could be chosen for performing classification in online or streaming environments. Nonetheless, no evaluations were performed varying the selected algorithm.

One of the problems in real-time applications is the limited availability of labelled data, not only because it might be difficult to collect a full training set, but also because instances could arrive sequentially (Wang *et al.*, 2014). This problem worsens in highly dynamic contexts, where new classes might appear and others could disappear. Thereby, techniques requiring a few or none labelled training instances are preferred. In this context, Sedhai and Sun (2018) proposed a semi-supervised tweet spam detection framework, comprising both online and batch phases. The online phase aims at operating in real-time and includes four spam detectors. First, a blacklisted domain detector that labels tweets containing blacklisted URLs. Second, a near-duplicate detector that labels tweets that are highly similar to pre-labelled tweets according to a MinHash clustering (Broder, 1997). Existing clusters are labelled according to a logistic classifier based on features representing the collective information obtained from all the tweets in them. Third, a ham detector that labels tweets that do not contain spam words and are posted by trusted users. Fourth, NB, logistic regression and Random Forest classifiers that label the remaining tweets based on hashtag, content, user and domain-based features. For a tweet to be considered spam, the three classifiers must label it as spam. Then, during the batch phase, the detectors are updated based on the labelled tweets from the previous time windows. According to the authors, updating the models allows the framework to capture new vocabulary and new spamming behaviours, making it capable of dealing with the dynamic nature of spamming activities. Experimental evaluation was based on the HSpam14 dataset (Sedhai & Sun, 2015), comprising more than 14 million tweets. Results showed the superiority of the framework when compared to the traditional NB, logistic regression and Random Forest classifiers. Small differences were observed when considering the batch model update. The presented framework showed more stable results as new tweets appeared, hinting its capabilities for adapting to new behaviours. Nonetheless, Random Forest achieved competitive results for several time-windows.

3.3.2 Knowledge-based techniques

Considering the brevity of social media texts, their high-dimensional feature space and the low term frequencies, the traditional BOW might not be the most appropriate model for representing short-text, as it might not preserve the semantic meaning of the original texts. Moreover, in social media, texts can include abbreviations, new terms and spelling mistakes, which worsens the synonymy, polysemy and normalisation problems. In this context, one possible solution for handling sparsity is to expand short-texts by adding new features based on semantic information extracted from external lexical databases. Following the trend, Yang *et al.* (2013) and Zhang and Zhong (2016) aimed at learning vector representations of both words and topics to improve short-text classification. Yang *et al.* (2013) proposed to combine lexical and semantic features for short-text classification. Semantic features are extracted from external knowledge repositories covering the domain of target classes. After the representation of topics is obtained from the external repositories (the authors chose *Wikipedia*) and the discriminative feature words for each topic are selected, each word in the short-text is assigned to the learned topics using a Gibbs sampling method. This allows to represent short-texts only using the mapped topics. Once words are assigned to topics, the semantic representations of short-texts can be built by replacing the original words with the corresponding topics. Thereby, short-texts are represented with a vector, in which each element is defined as the times the words in the short-text were assigned to the corresponding topic. These final representations are used for training an SVM classifier. Experimental evaluation based on short website snippets extracted from Google snippets and scientific abstracts in Ohsumed showed that the presented approach did not improve the results of considering LDA for learning the semantics of short-texts. The results of combining LDA with the presented approach slightly outperformed those of considering LDA. As the selected datasets might present different textual characteristics than noisy social media texts, the applicability of the approach on such environment cannot be guaranteed.

Zhang and Zhong (2016) aimed at learning vector representations of words and topics. This idea is similar to topical word embeddings, in which topics are regarded as pseudo-words to predict contextual words, with the difference that the authors decided to consider topics as new words that are added to the text. According to the authors, this allows to overcome the data sparseness problem of short-texts and makes possible to learn the vector representations of words and topics together. The proposed approach comprises three steps. First, topic learning with LDA from the external corpus, which results in the topic model for text inference and topic assignment of words. The topic assigned to each word is used to enrich the short-texts. Second, word/topic vector learning based on several variations of continuous BOW and skip-grams. Third, the obtained vectors are used for training a linear SVM. Experimental evaluation was based on a collection of Web search snippets, which according to the authors are short, sparse, noisy, and not topic focused. *Wikipedia* was selected as the external corpus. Several baselines were selected including the BOW representation, LDA topic extraction with a maximum entropy or SVM classifiers, link analysis in which short-text were enriched with the words associated with the most similar topic, and the approach in (Wang *et al.*, 2016a). Results showed that the proposed approach outperformed all baselines. The authors hypothesised that the poor performance of BOW was due to the low co-occurrence of terms, whilst the performance of LDA could be because its emphasis is on modelling topics and not word meanings. Link analysis was the best performing baseline, with differences of 1% in favour of the presented approach. The authors made no mention of using the approach on a real-time setting, nor updating the trained model, which hinders its applicability on social media environments.

As previously mentioned, short-texts are characterised by their brevity, noise and lack of contextual information, hindering any analysis based on statistics, and causing difficulties in the identification of senses for ambiguous words. These problems worsen in stream environments, in which short-texts continuously appear, accentuating the concept drift. Li *et al.* (2018) claimed that the characteristics of short-texts make difficult the application or adaptation of traditional text classification techniques. As a result, they proposed a promising stream classification approach based on external knowledge sources, concept clustering and incremental ensemble classifiers. The approach comprises several steps, which are executed for each set of instances that arrives in the stream. First, short-texts are analyzed and a backward maximum matching method is applied to efficiently find all terms matching concepts in the selected external knowledge source. Unlike most works that choose *Wikipedia* as the external source, the authors chose Probase (Wu *et al.*, 2012), due to the higher number of hypernym–hyponym relations between concepts. Second, once concepts are discovered, a disambiguation strategy is applied to reduce the impact of irrelevant senses. To that end, an entropy-based clustering is applied. To determine the dominant sense of an ambiguous term in a given short-text, the similarity between each concept cluster associated with the term and the concept cluster of all unambiguous terms is computed. After disambiguation, the selected concepts are clustered according to the senses selected for the terms to represent the feature space of short-texts. Each cluster indicates a sense represented by the discovered concepts. In this regard, a set of short-texts can be represented as the sense distributions of concepts. Third, a cluster-based topic drifting detection method is applied, in which the divergence between the sense distributions in two consecutive sets is analyzed to detect the concept drifts. Fourth, classifiers are built based on the defined feature space. One classifier is built for each set of short-texts. Finally, an ensemble classifier is built considering the individual classifiers built. When a new short-text to classify appears, it is represented in the defined feature space. Then, the k-nearest concept clusters are found by computing the semantic distance between the short-text to classify and the centres of the concept clusters in the recent k sets of short-texts. Once they are found, the short-text is assigned to the label of maximum probability. Experimental evaluation was based on four datasets including Web search snippets, short news and tweets. To study the performance of the approach in a real data environment, the authors simulated a stream of tweets sorted according to their timestamps. In the simulation, the authors included gradual and abrupt concept drifts, and noise. The selected baselines included both state-of-the-art concept drift detectors and several short-text classification approaches using LDA and *Wikipedia* as the external knowledge source. Results showed that the presented approach outperformed all concept-drift baselines, closely followed by an approach combining k-NN with probabilistic adaptive windowing. The highest differences were observed in the simulated streaming scenario, showing the capabilities of the presented approach. Similar tendencies were observed when compared with the traditional techniques for the news and snipped datasets. On the other hand, for the *Twitter* datasets, the presented approach did not achieved the best results. Particularly, the best results were obtained when considering LDA and a maximum entropy classifier, with differences up to 3%. This could imply the necessity of continuing the exploration of semantics of social media short-texts. In all cases, reported differences were statistically significant. Finally, the authors compared the computational complexity of the approach and the selected baselines. According to them, the baselines yielded a higher theoretical computational complexity than the presented approach. Nonetheless, the execution time comparison showed that under certain circumstances other techniques yielded similar or lower execution times.

Other potential use of external knowledge is to describe the classes in the absence (or limited availability) of labelled examples. In this context, Romero *et al.* (2013) and Li *et al.* (2017) proposed classification approaches requiring few or none labelled training data and leveraging on external knowledge sources. Romero *et al.* (2013) proposed a classification approach that only requires the definition of classes by means of ontologies or thesauri, independently of the existence of labelled data. Hence, whilst it does not depend on the frequency of examples of a class, it depends on the definition of that class and how the texts to be classified fit in such definitions. Therefore, the approach is appropriate for short-text classification scenarios where the frequency of occurrence of each class is small or even zero as no training examples nor training processes are required. Furthermore, the ontology of concepts can be applied to different document collections without extra efforts. The first step of the approach involves the definition of each class by adding a set of concepts that are semantically close to the name of the class, which are extracted from lexical or semantic databases, controlled vocabularies or thesauri. When possible, the reflexive, symmetric and transitive closures are computed for the corresponding semantic relations. Then, the compatibility degree between a text and each category is computed. The classes with the higher compatibility degrees are selected. Nonetheless, a text can also remain unclassified if no class is selected. Experimental evaluation was based on four datasets with news texts and Web snippets, which comprised less than 200 characters. All texts were pre-processed by removing stopwords and performing stemming over the remaining terms. The performance of the approach was evaluated in terms of precision, recall and F-measure. A text was considered to be correctly classified if at least one of the assigned classes matched the class assigned by a human expert. The performance of several semantic relations between concepts for determining the compatibility degree between a text and the classes was analyzed. Those relations included the usage of *ConceptNet*, *WordNet*, *Wikipedia*, *YAGO* and syntactic equality (only based on the class name). On average, the best results were obtained when selecting *Wikipedia* and *WordNet* as the semantic information sources. On the contrary, the worst results were obtained when considering only the name of the class, and the structural analogy and contextual neighbourhood extracted from *ConceptNet*. Results showed the importance of correctly and accurately defining the classes, as an inaccurate definition might lead to a high percentage of incorrectly classified or unclassified texts. The authors concluded that the effectiveness of the approach depended on a good pairing of the problem and the domain knowledge. The authors also highlighted that improvements on the similarity degree formula and on the ontology construction could lead to improvements on the accuracy results. Even though the

https://doi.org/10.1017/S0269888919000018

approach seems promising, it can only detect well-known classes for which representations have been curated. On online settings in which new topics to classify might constantly appear, it might be necessary to develop a mechanism for automatically generating their descriptions. This leads to the difficulty of selecting an adequate ontology that could cover all potential topics.

Unlike Romero *et al.* (2013) that did not require labelled data, Li *et al.* (2017) required a small labelled training set in a semi-supervised setting, which was then enriched with unlabelled short-texts. The authors presented a semi-supervised iterative classification algorithm based on fusion similarity and class centres. The approach first represents each class by selecting a set of words from the labelled training set based on their expected cross entropy. These class representations are used as prior knowledge to guide the classification process of unlabelled short-texts. Then, unlabelled texts are iteratively assigned to the most similar class definition according to a similarity function. The authors proposed to compute similarity based on combining the cosine theorem and the defined class representations. In each iterative step, one short-text is classified for each class. Then, class definitions are updated. The process stops when all unlabelled short-texts are classified. Experimental evaluation was based on approximately 8000 news titles posted from July to August 2015 belonging to eight classes. The presented algorithm was compared to k-NN and a state-of-the-art algorithm. According to the authors, their algorithm outperformed the selected baselines. Given the characteristics of incremental learning, the approach could be in theory applied in online settings. Nonetheless, the applicability of the approach in those environments in terms of scalability and the appearance of new classes remains to be evaluated.

3.3.3 Neural networks-based techniques

Traditional BOW models might have difficulties for capturing the semantic meaning of short-texts. To overcome this problem, several works (Wang *et al.*, 2016a; Dai *et al.*, 2017; Ravi & Kozareva, 2018; Yan *et al.*, 2018) have leveraged on word embedding models and deep learning techniques for short-text classification. Word embeddings aim at quantifying the semantic similarity of linguistic items based on the distributional properties of words in large textual samples. They allow to learn the semantic information of words and are one of the strongest trends in natural language processing. In this context, Dai *et al.* (2017) presented a clustering method based on word embeddings for health-related tweet classification. Tweets are represented based on word embeddings and then divided into clusters of words by means of a Dirichlet process. Finally, based on similarity measures of all clusters, tweets can be classified as related or unrelated to a topic according to a pre-defined similarity threshold. Experimental evaluation was based on 2,000 manually labelled tweets either related or unrelated to flu outbreaks. Word embeddings were trained with Google data. Results showed that the presented approach outperformed a simple NB only in the 33% of evaluations. According to the reported results, the approach showed to be sensitive to the similarity threshold, which caused a great variability of precision and recall scores. Interestingly, only two evaluations achieved both good precision and recall scores. Although the approach seems interesting, more evaluation and statistical analyses are needed to fully assess its benefits.

Also based on word embeddings, Wang *et al.* (2016a) proposed the definition of semantic clusterings for short-text classification, in combination with a Convolutional Neural Network (CNN) architecture. The goal was to introduce extra knowledge by pre-trained word embeddings to fully exploit the contextual information of short-texts. First, a clustering algorithm is used to cluster word embeddings and discover semantic cliques. Generally, the semantic meaning of short-texts is determined by a few key phrases that might appear at any part of the text. Hence, simply considering all words in the texts might introduce unnecessary divergence, hindering its semantic representation. To overcome this problem, the discovered cliques are used to extract semantic units (i.e., n-grams with a dominant meaning) from the short-texts, and capture their salient local information. Varying the sizes of the selected n-grams allows to exploit multi-scale contextual information, which could be helpful for reducing the impact of ambiguous words. The inputs for the CNN are the semantic units that yield a certain Euclidean similarity to the semantic cliques. Finally, the probability distribution is computed by a softmax layer. Experimental evaluation was based on short website snippets extracted from Google snippets and questions from the Text REtrieval Conference (TREC) dataset. The evaluation considered several baselines, ranging from an SVM trained

https://doi.org/10.1017/S0269888919000018

with textual features, their Part-Of-Speech (POS) tagging and hypernyms, variations of CNN based on pre-trained word embeddings and short-text enrichment by means of LDA. According to the authors, results showed that their approach outperformed every selected dataset. Interestingly, the best results for each dataset were obtained using different pre-trained word embeddings. Additionally, given the intrinsic computational complexity of deep learning approaches, the authors should have included a comparison of execution times or complexity of the compared approaches. The authors made no mention to using the approach on a real-time setting.

Recent success achieved by deep learning approaches relies on the existence of large amounts of labelled instances. Given the sparsity and brevity of short-texts, each text might not provide enough information for labelling it or discriminating it in the feature space. As a result, thousands of labelled examples might not be sufficient for classifiers to obtain satisfying performance (Nakov *et al.*, 2016) In this context, Yan *et al.* (2018) presented a sentiment classification technique relying on a limited number of labelled instances and metric learning based on deep neural features and Siamese CNN. The framework comprises two training steps; first, the pre-training of a CNN model, and second, the training of a Siamese CNN based on the pre-trained CNN model and fine-tuned with the small number of labelled instances. At the beginning of the pre-training, sentences should be tokenised and converted into word vectors by applying word embeddings. In this case, the authors chose word2vec. During the pre-train, the CNN parameters are initialised. The chosen CNNs are optimised for pairs of text matrices and represent the inputs by a nonlinear mapping. In this step, the selected softmax classifier is not fine-tuned as the dataset is unbalanced and the classifier tends to identify all samples as the major class. During the second training, the sentences with different topics and typical structure are regarded as prototypes, regardless of whether their classes match. Additionally, the samples that the pre-trained softmax classifier fails to identify are also included in the prototype set. Then, based on a stochastic sampling strategy, the prototype is combined with a predefined number of samples, which are used for training the Siamese CNNs. According to the authors, the Siamese CNNs should be given a 1 : 1 ratio of same-different class to train on, to eliminate the negative influence of imbalance. In this step, the softmax classifier is replaced by a cost-sensitive SVM classifier, which is less biased to unbalanced class distributions. Experimental evaluation was based on four *Twitter* datasets (Multigames, Health Care Reform, the SentiStrenght dataset and the dataset from SemEval-2013). The performance of the presented technique was compared to traditional classification techniques (such as Adaboost, SVM, Maximum Entropy, RF and NB), and two types of deep learning methods (recurrent neural networks—RNN and long short-term memory—LSTM) with BOW models. For the traditional classification techniques, tweets were pre-processed by removing stopwords, performing POS tagging and using Sentiment WordNet. To weaken the dataset unbalance, the authors applied random down-sampling. Results showed that the presented approach outperformed all baselines in terms of accuracy. The authors hypothesised that this could be caused by the unbalanced nature of datasets. Given that no statistical analysis of results was performed, it cannot be guaranteed that the observed differences are not due to chance. Moreover, considering the dataset unbalance, accuracy might not have been the most adequate choice for analyzing the performance of classifications. Although the authors expressed their interest on classifying social media data, they did not mention the possibility of updating the trained models as new instances are known, nor performed an analysis of computational complexity and scalability. In summary, even though the approach seems promising, more evaluations and comparisons are needed to effectively assess its capabilities.

Although neural networks are widely used, they tend to require intensive resources, both for computation and for memory/storage performance. Thus, one of the biggest challenges is to efficiently run these complex networks on mobile devices with tiny memory footprint and low computational capacity. Ravi and Kozareva (2018) aimed at tackling this challenge for short-text classification in the context of dialogue act identification. To that end, they proposed a self-governing neural network (SGNN), which can learn compact projection vectors with local sensitive hashing. The self-governing property stems from the ability of the network to learn a model without incurring in the cost of initialising, loading or storing any feature or vocabulary weight matrices. In this sense, this model, unlike the majority of the widely used state-of-the-art deep learning techniques, is embedding-free. The hashing is used to reduce the input

https://doi.org/10.1017/S0269888919000018

dimensions to a short fixed-length sequence of bits, which allows saving storage and computational cost. Then, the projection of an incoming text can be computed with a small memory footprint on the device. According to the authors, the advantage of SGNNs over other networks is that they do not need pre-trained word embeddings nor a high number of parameters. Experimental evaluation was based on two dialogue act benchmark datasets. No pre-processing was applied to the selected datasets. Improvements over simple baselines such as assigning the majority class and a NB classifier ranged between 12% and 36% in terms of accuracy. The proposed network was also compared to state-of-the-art techniques such as CNN and RNN variants. Results showed that SGNN was able to outperform every technique but RNN for one of the datasets. Although the authors described the computational complexity of the approach, they did not present a time complexity comparison with the other baseline techniques. Moreover, the authors did not describe the mobile devices on which evaluations were performed.

3.4 Summary

The problem of classification has been widely studied in several fields, such as data mining, machine learning and information retrieval, amongst others. Text represents a specific kind of data in which word attributes are sparse and highly dimensional, and frequencies are low for most words (Aggarwal & Zhai, 2012). Almost all known learning techniques have been used to effectively classify text data (Sebastiani, 2002). However, with the advent of short-texts and social media data, new challenges in the automatic learning process need to be faced, especially in resource-constrained scenarios (Aggarwal, 2014). This lead to the development of new techniques, which are summarised in Table 1, based on the type of features used, whether they are presented for batch or online environments, and how evaluation was performed, amongst other relevant characteristics. As previously mentioned, new techniques should aim at tackling the continuous appearance of new short-texts, the sparsity of the feature space, the lack of contextual or semantic information, and the lack of a grammatical structure, amongst other problems. Nonetheless, most of them share the same drawbacks. First, they are mostly based on textual information, neglecting other sources of heterogeneous information provided by social networking sites. Second, most of the techniques do not explicitly provide mechanisms for updating the trained models. Additionally, most approaches lack a systematic evaluation including statistical analyses of the observed differences, which would allow to fully assess and compare the performance of the different techniques.

4 Clustering

Due to the increasing availability of data, human labelling has become a difficult and expensive task (Alelyani *et al.*, 2013). Unlike classification, clustering refers to the problem of grouping instances into groups or clusters (i.e., the unsupervised analogous for classes or categories) without advance knowledge of the class or category they belong to. Usually, neither the description nor quantity of clusters are known in advance unless there exists domain knowledge, which poses a great challenge for data clustering. Figure 2 depicts the general clustering process (Jain & Dubes, 1988; Xu & Wunsch, 2005). As with the classification task, there is a set of instances comprising features. However, unlike for the classification task, instances are not labelled. A feature selection process might be applied to restrict the set of features to be used. Then, instances are used to build a clustering model. The selection of the clustering algorithm usually involves choosing a similarity or proximity metric that directly affects the resulting clusters. Next, the obtained clusters are validated. Clustering algorithms can always generate a division, regardless of whether such structure actually exists. Different clustering algorithms usually lead to different cluster divisions. Moreover, for the same algorithm, parameter tuning or input order might also affect the final cluster division. Thereby, effective evaluation standards and criteria are important to assess the performance and confidence of the discovered clusters. Traditionally, there are three categories of evaluation criteria (Jain & Dubes, 1988; Xu & Wunsch, 2005): external, internal and relative indexes. External indexes are based on a pre-defined structure that reflects prior information on the data, which is used as standard. On the contrary, internal indexes are independent from external information as they examine the clustering structure directly from the original data. Relative criteria emphasises the comparison of

https://doi.org/10.1017/S0269888919000018

Table 1 Summary of classification techniques

	Task	Critical time	Binary or multi-class?	Updates over time?	Computational complexity	Type of features	Includes pre-processing?	Evaluation
Yuan *et al.* (2012)	Text classification	Batch	Unknown	No	Low	Textual	Stopword and low frequency word removal. Stemming	Over than 3 million questions from *Yahoo! Webscope* dataset
Kim *et al.* (2014)	Text classification	Batch	Multi-class	No	Medium	Textual	Unknown	English (ODP) and Korean (Daum directory) datasets
Su-zhi and Pei-feng (2011)	Text classification	Batch	Multi-class	No	Medium to high	Textual	Unknown	10 567 comments extracted from TianYa BBS, Sohu and Sina News
Cui *et al.* (2016)	Text classification	Batch	Multi-class	No	Medium	Textual	Username and redundant punctuation removal, link tokenisation	Over 175 000 tweets collected from normal consumers mentioning certain brands
Collins *et al.* (2015)	Sentiment analysis	Batch	Multi-class	No	Medium	Textual	Tokenisation, lower casing and URL replacement	Dataset in SemEval-2015
Nishida *et al.* (2011)	Text classification	Batch	Multi-class	No	Medium	Textual	Unknown	Over than 1 000 000 tweets
Ramírez de la Rosa *et al.* (2013)	Text classification	Batch	Multi-class	No	Medium	Textual	Unknown	Reuters-21578
Sedhai and Sun (2018)	Semi-supervised spam detection	Online	Binary	Yes	Low to medium	Textual, hashtags, user and domain features	No	HSspam14
Yang *et al.* (2013)	Text classification	Batch	Multi-class	No	Medium to high	Textual	No	Website snippets from Google snippets and scientific abstracts in Ohsumed

Zhang and Zhong (2016)	Text classification	Batch	Multi-class	No	Medium	Textual	No	A collection of Web search snippets
Li *et al.* (2018)	Text classification	Online	Multi-class	Yes	Medium	Textual	No	OFour datasets including Web search snippets, short news and tweets
Romero *et al.* (2013)	Text classification	Batch	Multi-class	No	Low	Textual	Stopword removal and stemming	Four datasets with news texts and Web snippets comprising less than 200 characters
Li *et al.* (2017)	Semi-supervised text classification	Batch	Multi-class	Yes	Low to medium	Textual	Title segmentation and stopword removal	8 000 news titles
Dai *et al.* (2017)	Text classification	Batch	Binary	No	Medium	Textual	Stopword removal and non-specified others	2 000 manually labelled tweets related to flu outbreaks
Wang *et al.* (2016a)	Text classification	Batch	Multi-class	No	High	Textual	No	Short website snippets from Google Snippets and questions from the TREC dataset
Yan *et al.* (2018)	Text classification	Batch	Multi-class	No	High	Textual	No	Four *Twitter* datasets: Multigames, Health Care Reform, the SentiStrenght dataset and SemEval-2013
Ravi and Kozareva (2018)	Dialogue act classification	Online	Multi-class	No	Low to medium	Textual	No	Two dialogue act benchmark datasets

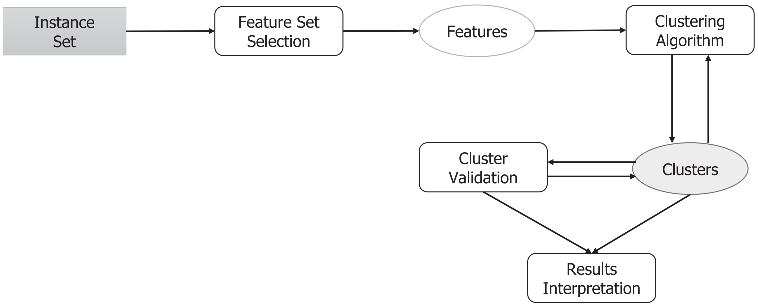

Figure 2 Overview of the clustering process

different clustering structures to decide which one might more accurately represent the characteristics of the original instances. Finally, results are interpreted. The final goal of clustering is to provide meaningful insights from the original data.

4.1 Traditional clustering techniques

Traditional text clustering techniques compute the similarity between two documents heavily relaying on the co-occurrence of terms between them. Due to the low frequency of terms appearing in short-texts, such metrics might not be suitable for assessing short-text similarity (Ni *et al.*, 2011). As a result, traditional clustering techniques might not achieve high performance when applied to short-texts. Several authors (Kang *et al.*, 2010; Rangrej *et al.*, 2011) compared the performance of clustering algorithms that are traditionally applied to long-texts for the task of short-text clustering. Rangrej *et al.* (2011) evaluated the performance of three traditional algorithms: k-Means, singular value decomposition (SVD) and the graph-based approach affinity propagation (AP). The authors also evaluated the performance of Jaccard and Cosine similarity metrics. Experimental evaluation was based on a small number of tweets. IDF vectors were obtained from tweets after removing stopwords and performing stemming. The best results were achieved with the graph-based approach, whereas the worst results were obtained with SVD. Regarding the evaluated similarity metrics, results varied according to the clustering algorithm used. Whilst k-Means achieved the best results when considering Cosine similarity, the graph-based approach achieved the best results when considering Jaccard similarity. Finally, the authors also studied the effect of cluster overlapping (e.g., tweets might be associated with two different topics) in the SVD algorithm. The performance worsened when tweets were assigned to multiple clusters, which could be caused by the short nature of tweets, and thus the difficulty of identifying multiple overlapping topics.

Kang *et al.* (2010) proposed to apply AP for clustering tweets that contained links to news articles based on the supposition that tweets sharing the same link belonged to the same cluster. The authors only considered content-based features extracted from tweets such as: words, hashtags and links, and the Cosine similarity between them. Experimental evaluation was based on data extracted from *Tweetmeme*,[8] a news platform that no longer exists. The dataset comprised 398 stories originally posted between June and July 2009 and their 1000 latest re-tweets. Tweets were processed by applying stemming and lower-casing, and discarding non-word characters. The remaining words were weighted using TF-IDF. As the dataset comprised tweets with similar sets of words, the authors chose to collapse all identical or similar tweets into only one text to reduce the dataset size, and thus the noise. The performance of AP was compared to the golden standard, which considered the tweets URLs as the actual cluster labels. The authors stated that the erroneous clustering could be because tweets belonging to different clusters might share the same hashtags, or tweets with the same link might have different hashtags. As a result, more complex similarity metrics should be introduced to improve the news detection approach; for example, similarities could consider the time-stamp of tweets or the relevance of users posting the tweets.

[8] http://tweetmeme.com/.

In summary, although the reviewed techniques obtained promising results, their applicability in real-time might be hindered by their computational complexity. As a result, they might present scalability issues when evaluated with large-scale datasets. Furthermore, short-text clustering results should be compared to long-text results in order to correctly assess the capabilities of traditional algorithms to cluster short-texts.

4.2 *Enhancements to traditional clustering techniques*

Finally, several works (Tu & Ding, 2012; Li *et al.*, 2012; Ferrara *et al.*, 2013; Parikh & Karlapalem, 2013; Zhang *et al.*, 2014; Wang *et al.*, 2016b) aimed at enhancing traditional algorithms to improve their performance for short-text clustering. Particularly, the approaches focused on designing new similarity metrics. Zhang *et al.* (2014) proposed a similarity metric (*RepSim*) that instead of assessing the explicit content of posts analyzes the re-posting behaviour of users as a representation of their interests. In consequence, if two posts are re-posted by the same user, they are likely to be more similar than two randomly selected posts. In this regard, *RepSim* defines the degree of similarity between two posts as the ratio of users who have re-posted them. Experimental evaluation was based on a large-scale training set, and a small test set extracted from Sina-Weibo. Clustering was performed by means of k-Means++ with initial centres artificially selected according to the training set distribution. *RepSim* was compared to the Cosine similarity between the TF-IDF values. Results showed that *RepSim* was able to outperform content-based similarity functions. According to the authors, *RepSim* has several advantages regarding other clustering algorithms. First, it can be used in online and real-time environments, as it is not computationally complex. Second, it depends on the behaviour of users instead on the content of posts, which could help to overcome the sparseness problem posed by short-texts. However, the generalisation of results for bigger and probably multiple test sets with diverse degrees of overlapping with the training set remains to be validated.

Tu and Ding (2012) proposed a semi-supervised modification to the SinglePass clustering algorithm for event detection specially designed for micro-blogging data. The event detection process consists of two steps. First, a NB classifier is used to discard tweets referring to the daily life of users. Then, the remaining texts are clustered. Considering that the SinglePass algorithm is highly influenced by input order, the authors proposed to sort the input according to the number of repeated features and their mean weight. The authors assumed that if many texts are related, features would be repeated and, thus, a text with many repeated features would be useful to start the clustering process. Experimental evaluation was based on 10 590 short-texts extracted from Sina-Weibo. Features from short-texts were weighted using TF-IDF, assigning higher weights to nouns and verbs. Results showed that the approach outperformed the original SinglePass algorithm, thus representing an efficient and accurate alternative to detect topics in a micro-blogging environment.

Similar to Dai *et al.* (2017), Wang *et al.* (2016b) aimed at representing short-texts as distributed vectors with neural networks for semi-supervised clustering. Their goal was to combine the representation learning process and k-Means to simultaneously update both. During the training phase, neural networks and centroids are randomly initialised. Then, cluster centroids, the cluster assignments for each text and the parameters within the deep neural network are iteratively optimised through three steps until convergence. First, each short-text is assigned to its nearest centroid based on its representation from the current neural network. Second, cluster centroids are re-estimated based on the newly clustered text. Third, neural networks are updated by keeping the centroids and cluster assignments fixed. The objective function comprised two terms. The first is an adaptation of the k-Means algorithm, whilst the second encourages labelled data to be clustered in correlation with the given class labels. The performance of two types of neural networks (CNN and LSTM) for generating the text representations was analyzed. Experimental evaluation was based on four labelled datasets comprising questions, short news and an ontology. Different text representations (e.g., BOW and TF-IDF weighted BOW) in combination with k-Means were used as baselines, as well as adaptations of neural networks for classification. The adaptation consisted on adding an extra layer to the networks used for creating the text representations. Results showed that the classification results outperformed the clustering ones, implying that a small amount of

https://doi.org/10.1017/S0269888919000018

labelled data is necessary for increasing performance. Moreover, the best performing alternative was the CNN variation of the presented approach for most datasets, showing the power of deep learning models for short-text modelling.

Li *et al.* (2012) proposed a new similarity metric based not only on the co-occurrence of terms of tweets, but also on the frequency pattern of terms in a particular time window. The metric computes the similarity between two tweets in a particular time window by dividing such time window into sub-windows and then multiplying the scores for each sub-window. Content similarity is defined as the Cosine similarity between the tweets in that particular sub-window. Term frequency is assessed by considering the normalised frequency of each element in the sub-window, with regard to the frequency in the original time window. Then, the final similarity score is computed by adding the scores obtained for each sub-window. As a result, two tweets are similar if they share similar content and their frequency patterns are consistent along the time windows. Tweets either having inconsistent frequency patterns or dissimilar content are regarded as dissimilar. Dissimilar content might suggest that the compared tweets refer to different events. On the other hand, similar content in combination with inconsistent frequency patterns might suggest that the compared tweets refer to similar events that occurred at different time periods. The authors chose to apply their similarity metric on a Nearest Neighbour clustering algorithm, setting the number of nearest neighbours to three. The performance of the proposed similarity metric was compared to that of Event Detection with Clustering of Wavelet-based Signals (EdCoW) (Weng & Lee, 2011) and a variant of the presented metric based on unigrams. Results showed that the proposed metric was able to improve the precision of both. On the other hand, the highest recall was observed when considering unigrams. In consequence, tweet representation is also an important factor to consider when designing or testing a similarity metric as it ultimately affects the performance of the clustering approach. Nonetheless, although the authors reported improvements of their approach regarding other techniques, as they combined tweet segmentation and the new similarity metric, the source of the performance improvements cannot be accurately determined. The authors planned to extend the similarity metric to consider additional tweet features, such as re-tweeting or hashtag patterns.

Finally, other works (Ferrara *et al.*, 2013; Parikh & Karlapalem, 2013) focused on the definition of alternatives to assess the textual similarity of short-texts. Considering that *Twitter* enforces a maximum number of characters per tweet, Ferrara *et al.* (2013) stated that tweets should not be considered separately. Instead, they should be clustered in units of information, ideas or concepts (i.e., memes) that could be spread through social networks. Alternatively, the unit of information can be defined as a tweet set, also called ‘protomeme’, carrying the same piece of information or entities, such as hashtags, mentions, URLs or textual phrases. Those similarity measures were based on textual content, tweets metadata, network features and their combinations. Clustering was intended to be an asynchronous and offline process that at pre-determined time intervals analyzes and clusters the protomemes obtained in a recent time interval. Four protomeme similarity metrics were proposed based on the Cosine similarity: common user (similarity between the number of times users have used the protomeme), common tweet (similarity between the tweets included in the protomeme), content (TF-IDF similarity between the tweets comprising the protomeme) and diffusion similarity (similarity between the users that had posted, mentioned or re-tweeted a tweet included in the protomeme). Additionally, those metrics are aggregated using pairwise maximisation and linear combination. The pairwise maximisation aimed at choosing the metric that obtained the highest value for two instances. The rationale behind this strategy was to capture the greatest relatedness of each particular pair of instances, as for example, the relatedness of two instances might be better described by their content, whereas other pair of instances might be better described by their users’ activity. The second alternative aimed at computing a weighted average of all similarities, requiring a non-trivial computation of the optimal weights. Experimental evaluation was based on a medium-scale number of tweets regarding the US presidential primaries in 2012. Tweets were manually labelled by identifying topics comprising at least three tweets. The evaluation aimed at assessing whether protomemes could help to identify news in social media. To that end, the authors compared the proposed similarity metrics to two baselines: a content-based similarity based on TF-IDF and a combination of network-based features and content. The pairwise combination of similarities achieved the best performance for numbers of clusters close to the ground truth, outperforming the network-based baseline.

Additionally, the similarity pairwise maximisation was as effective as the non-trivial parameter optimisation of the linear combination. The authors concluded that the similarity metrics in combination with the concept of protomeme provided significant enhancements to the task of news detection in the context of a fully automatic, unsupervised and scalable approach.

Parikh and Karlapalem (2013) proposed a similarity metric based on a linear combination of the Jaccard similarity of both content and frequency patterns. Unlike the approach presented by Li *et al.* (2012), the authors claimed that elements having similar frequencies in every time interval do not represent an event, and thus could be discarded. As a result, the similarity score was only computed for those elements showing an increment in their appearance frequency in any of the considered time intervals. Then, the similarity metric was inserted into an agglomerative hierarchical clustering algorithm. Experimental evaluation was based on a large-scale tweet dataset related to the broadcasting domain. Although the authors claimed to have presented a scalable and efficient system that achieved high precision, they mistook the definitions of performance metrics (precision and recall) claiming that recall results cannot be compared across datasets, whereas precision results can. Moreover, the authors did not define a baseline for comparing the obtained results. Consequently, further evaluations are needed to truly assess the efficiency of the similarity metric.

4.3 New clustering techniques

In contrast to the previously described approaches, several authors (Yang & Ng, 2009; Carullo *et al.*, 2009; Tsur *et al.*, 2013; Kim *et al.*, 2013) argued that traditional text clustering techniques that have good precision and recall when applied to long-texts usually present a poor performance when applied to short-texts. The intrinsic characteristics of short-text pose a new challenge to clustering approaches since a massive amount of texts (posts, tweets, etc.) is available, but a large proportion of it is meaningless. In this regard, novel clustering techniques have been proposed to deal with this problem and the characteristics of short-texts. The techniques presented in this section are organised into four categories, according to how they tackled the classification task: word/character-based techniques, topic modelling-based techniques, partitioning-based techniques (including techniques based on similarity or distance metrics) and density-based techniques.

4.3.1 Word/character-based techniques

Hashtags are creative labels used in social media to characterise the topic of the associated short-texts. Regardless of the intended use, hashtags cannot be used to cluster texts with similar content (Stilo & Velardi, 2017). First, due to the spontaneous and highly dynamic way in which they are created by users in multiple languages. Thus, the same topic can be associated with multiple hashtags, and conversely, the same hashtag might refer to different topics in different time periods. Second, hashtags are more difficult to disambiguate than common words, as no sense catalogue is available. Third, hashtags might be difficult to analyze as they often consists of acronyms and concatenated words. The real time detection of related hashtags could be used to improve the task of hashtag recommendation, which could facilitate the monitoring of online discussions. In this context, Stilo and Velardi (2017) proposed a temporal sense modification of the Symbolic Aggregate ApproXimation (SAX) (Lin *et al.*, 2003) technique (named SAX*) based on temporal co-occurrence and similarity of the related time series. SAX* is based on the idea that hashtags with similar temporal behaviour are semantically related. The nature of this relatedness is either systematic and repetitive, connected with a specific event or is a synonymy relation. SAX* comprises four steps. First, the temporal series associated with hashtags are partitioned into sliding windows, normalised and converted into symbolic strings using the original SAX. This step has two parameters, the dimension of the alphabet and the number of partitions of equal length. Second, the temporal series of known keywords associated with events are converted to symbolic strings for automatically learning regular expressions representing common usage patterns of words in a compact way. Only active hashtags with frequencies higher than a pre-defined threshold, and matching one of the learned regular expressions are considered in the subsequent steps.

https://doi.org/10.1017/S0269888919000018

Third, hashtags are analyzed in sliding windows and active hashtags are clustered using a bottom-up hierarchical clustering algorithm. Fourth, clustering is used to cope with temporal collisions based on detected components in the graph formed by the hashtags in each detected cluster. The rationale behind this step is that if two hashtags do not sporadically co-occur, then the underlying senses must be related. Experimental evaluation was based on a tweet collection crawled during 3 years comprising approximately 5.1 million multilingual tweets. The performance of SAX* was compared to that of k-means lexical clustering. Results showed that under an appropriate parametrisation, SAX* could detect all events, including different concurrent ones. Moreover, results showed that k-means performed poorly even in small temporal windows when merging all tweets including the same hashtag. The computational complexity analysis showed that SAX* was more efficient in two orders of magnitude than LDA and k-means, which could make SAX* suitable for streaming environments. The authors concluded that the coverage of hashtag usage patterns was affected by the density of the analyzed stream. With sparse streams, world-wide events can be discovered, whilst minor events are either missed or conflated into the same cluster. The authors stated that precision and recall in those events could be significantly improved when considering denser streams, possibly by using the geolocalisation of texts to better separate synchronous events, though it remains to be evaluated.

Yin *et al.* (2018) also aimed at tackling the explosive growth, sparsity and concept drift of short-texts in social media by proposing a stream clustering technique based on the Dirichlet process multinomial mixture model (named MStream). The MStream technique assigns a short-text to either an existing cluster or a new one based on the probabilities computed by the Dirichlet model. In this way, MStream can detect new clusters more naturally and deal with the concept drift problem. Clusters are represented by a vector combining its short-texts into a vector containing the list of word frequencies, the number of short-texts and the number of words. The technique performs a one pass clustering process, and then applies an update clustering process on each batch. The one pass clustering process can be used to deal with the one pass scheme of text streams. On the other hand, the update clustering process can be used to deal with the batch scheme of text streams. When a new batch of short-texts comes, the one pass clustering process is used to obtain an initial clustering result with one iteration of the batch. Then, the update clustering process is used to update the clustering results. In the last iteration, each text is reassigned to the cluster with the highest probability. As the number of clusters increases with more texts appearing, the space and time complexity of MStream will grow too large if outdated clusters are not deleted. In this context, the authors proposed a modification of MStream with forgetting rules (named MStreamF), which can delete outdated short-texts by deleting clusters of outdated batches. Experimental evaluation was based on three datasets: tweets relevant to the queries in the TREC 2011-2015 tracks, a news titles dataset and a temporally sorted dataset combining the previous ones. Four baselines were selected including LDA and state-of-the-art streaming clustering techniques considering the time-varying distributions of topics. Results showed that MStream and MStreamF achieved the highest quality clusters. When comparing MStream and MStreamF, the authors determined that MStreamF performs better on datasets organised by topics, showing the importance of the forgetting strategy. A comparison of execution times showed that MStream and MStreamF with one pass were faster than the selected baselines. Moreover, MStreamF showed to be faster than MStream as it kept fewer clusters. As future work, the authors intend to use the proposed techniques in other related text learning applications, such as search result diversification, text summarisation and event detection and tracking.

4.3.2 Topic modelling-based techniques

Liang *et al.* (2016) aimed at tackling two of the challenges of short-text clustering. First, the change in topic distribution over time, as topics do not only emerge but also fade. Second, most works assume that the available content is rich enough to infer topic information per each document. However, short-texts are sparse and the number of words in each document is limited, thus hindering the accurate inference of topic distributions. To this end, the authors proposed a dynamic clustering topic (named DCT) model for tweet retrieval, which allows to track the time-varying distributions of both topics over documents and words over topics. DCT involves a dynamic Dirichlet multinomial mixture model (DDMM) that captures short- and long-term temporal dependencies and allows tracking the dynamic topic distributions over short-texts streams. The DDMM is complemented with a collapsed Gibbs sampling algorithm that infers the changes in topics and document probability distributions. This sampling assigns a single topic to all the words of a short-text, and then uses the already inferred topic distributions as a prior of the topic distribution of the current document, whilst allows new texts to change the posterior distribution of topics. Evaluation was performed over a *Twitter* corpus including more than 369 millions of manually tagged tweets. The evaluation aimed at inferring the relevance of each cluster in relation to a user query. DCT was compared with five baselines and state-of-the-art techniques: Language Model (ranks documents by their relevance computed based on a multinomial query likelihood model) (Croft *et al.*, 2010), Time-aware Microblog Search (first adopts a feedback framework where temporal features are extracted from an initial ranked list of texts, and then re-ranks the list to produce the final ranking) (Efron *et al.*, 2014), latent Dirichlet allocation, Dirichlet Multinomial Mixture Model (similar to DCT without the temporal dependencies) (Yin & Wang, 2014) and Topic Tracking Model (clusters documents based on a dynamic topic tracking model capturing the temporal dependencies between long-text streams) (Iwata *et al.*, 2009). Results showed that DCT significantly outperformed the other techniques. According to the authors, DCT was better at tracking changes of topics than Topic Tracking Model, as the latter focuses on long documents. One of the disadvantages of DCT is the need of manually defining the number of clusters, which the authors planned to tackle in future work. Additionally, they proposed to apply DCT in other text applications such as tweet summarisation and sentiment analysis.

Similar to Liang *et al.* (2016), Jia *et al.* (2018) aimed at tackling the sparsity and high-dimensionality of short-texts. The authors proposed a concept decomposition method (named *WordCom*) that creates concept vectors by identifying semantic word communities from a weighted word co-occurrence network extracted from a corpus. Then, cluster memberships are estimated by mapping the original short-texts to the learned semantic concept vectors. *WordCom* has four steps. First, a co-occurrence network is built from a corpus. Second, semantic word communities are extracted from the network by using a modified k-Means algorithm in which the initial cluster centres and the number of clusters are determined by selecting k potential centres such that they have a higher density than their neighbours and a relatively large distance from points with higher densities. Third, word communities and their corresponding centres are combined to form concept vectors. Fourth, short-texts are projected into the concept vectors to obtain their cluster memberships. Moreover, it is possible that the words in a small subset of a short-text corpus cover the majority of the words in the full corpus. Thereby, the concept centres can be obtained only from the selected subset, making the method easily scalable for large short-texts corpora. Experimental evaluation was based on two Chinese and three English datasets. The method was compared with existing state-of-the-art algorithms including heuristic optimisation methods *TermCut* (Ni *et al.*, 2011), k-Means, spherical k-Means, and LDA, amongst others. Results showed that as the shortness and sparseness of texts increased, the improvements of *WordCom* over the other algorithms also increased. Nonetheless, *WordCom* was shown to present difficulties when considering long-texts. The authors hypothesised that it could be caused by the heavily overlapping community structure of the resulting word co-occurrence network.

4.3.3 Partitioning-based techniques

In partitioning techniques, instances are represented as points, and a similarity or distance metric is defined between the pairs of points in the space. The goal of this type of techniques is to separate the points into clusters to maximise or minimise a given function based on the selected similarity or distance metric. Generally, partitioning techniques require the number of clusters to be pre-defined, which hinders their applicability in highly dynamic domains where such number is unknown. The problem of detecting topics in micro-blogging networks has been frequently cast to a clustering task (Carullo *et al.*, 2009; Ni *et al.*, 2011; Kim *et al.*, 2012; Tsur *et al.*, 2013; Popovici *et al.*, 2014). Kim *et al.* (2012) proposed the new Core-Topic-based Clustering (CTC) algorithm that aimed at simultaneously extracting topics and clustering tweets. The method is based on jointly minimising the inter-cluster similarity and maximising the intra-cluster similarity. CTC represents tweets in the form of a graph where each vertex indicates a

https://doi.org/10.1017/S0269888919000018

tweet, and the edges connecting them are weighted with their similarity. Each cluster is considered to be based on core topics, that is, proper nouns. Then, k clusters are generated by grouping all tweets having the corresponding topic. Clusters are evaluated considering the relation between inter- and intra-cluster similarity. Finally, the top clusters that maximise the computed relation boosted by the proportion of re-tweets contained in the cluster are selected. Experimental evaluation was based on a dataset including tweets regarding four popular TV shows. Tweets were pre-processed by only selecting proper nouns, sequences of words beginning with capital characters and phrases enclosed by quotation marks, which were weighted according to their TF-IDF score. Tweets were assigned to a category if they contained the associated name of the show or any of the related hashtags. The new approach was compared to k-Means. Results showed that CTC was more efficient than k-Means and that it produced more stable cluster distributions than k-Means. In summary, CTC was able to improve clustering results of k-Means and retrieve more meaningful topics than k-Means.

Ni *et al.* (2011) proposed a new short-text clustering algorithm (named *TermCut*) based on representing the short-texts as a graph and finding core-terms. Each node of the graph represents a short-text and the weighted edges represent the similarity between the connected nodes. The algorithm recursively derives the core-terms and bisects the graph in two sub-graphs: one comprising the texts that contain the core term, and the other comprising the texts that do not contain the core term. The criterion to find core-terms (RMcut) is also based on minimising the inter-cluster similarity, and maximising the intra-cluster similarity. RMcut tries to decrease the impact of the sparsity of short-texts by adding the similarities between all the texts within a cluster. Two alternative termination conditions were presented. The first strategy, Cluster Number based TermCut (CNTC), is based on creating a pre-defined number of clusters. On the other hand, the second strategy, Threshold based TermCut (TTC), is based on the fact that some applications do not have prior knowledge of the number of clusters to be created, and thus computes the difference between two consecutive iterations of the RMcut value for the cluster set. If such difference is higher than the pre-defined threshold, the process continues and the selected core-term is used to bisect its corresponding cluster. Otherwise, the process is terminated. Experimental evaluation was based on two datasets. The first comprised a small set of Chinese questions collected from the interactive question answering system *BuyAns*,[9] manually classified into 13 classes. The second dataset comprised a medium number of search snippets extracted from Phan *et al.* (2008) classified into eight classes. Stopword removal and stemming were applied to both datasets. The six clustering algorithms in the CLUTO[10] package were used as baselines. Results showed that CNTC outperformed the other algorithms for both datasets. The worst results were obtained for the smallest classes, showing that RMcut was not able to effectively find core-terms for them.

Unlike the previous techniques that require all instances to be known at the start of the processing phase, online algorithms are designed to process data as it becomes available. This characteristic makes them more suitable for online or streaming environments. In this regard, there are a number of techniques for online clustering proposed in the literature (Yang & Ng, 2009; Carullo *et al.*, 2009; Tsur *et al.*, 2013; Popovici *et al.*, 2014). Yang and Ng (2009) proposed a scalable and iterative distance-based clustering algorithm, named SDC. The technique ensures that initial clusters have a pre-defined density. Then, clusters are expanded by means of scalable distances. Additionally, SDC is supposed to filter noise, and it does not require to specify the number of clusters. Iteratively, a seed text is selected, and its set of neighbours is defined. For a text to be considered a neighbour, its distance to the seed has to be higher than a pre-defined threshold (*eps*). If seeds have a higher number of neighbours than a pre-defined threshold, a new cluster is created including both the seed and its neighbours. Otherwise, the seed text is marked as unclustered. The iterative process continues until all texts are analyzed. Then, the centroid of each initial cluster is computed. All unclustered texts whose distance to a centroid is larger than $eps = eps - \Delta_{eps}$ are added to the corresponding cluster, whereas texts that cannot be clustered are treated as noise. The distance between texts is measured by the Cosine distance. Experimental evaluation

[9] http://www.buyans.com/.

[10] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview.

was based on political threads extracted from *MySpace* from May 2008. Each thread was represented with the 20 words with the highest TF-IDF score. SDC was compared to the traditional Density-Based Spatial Clustering of Application with Noise (DBSCAN) (Sander *et al.*, 1998). Results confirmed the relation between the number of created clusters and the overall accuracy of the approach. SDC was shown to correctly identify topics, and thus to cluster together texts with similar content. The authors concluded that their findings ensured that a scalable distance-based approach is more suitable than a pure density-based approach for clustering short-texts.

Carullo *et al.* (2009) presented a technique (named *ArteCM*) based on adapting the number of created clusters to the provided data, and to use a BOW+POS representation of texts, which allows a similarity-based definition of cluster centroids. The technique is designed to scan the text collection only once. For each text to be classified, its nearest cluster (*Cm*) is identified. If the similarity between the text and *Cm* is higher than a pre-defined threshold ε, the text is recognised as part of *Cm*. Then, the new clustered text contributes to the cluster definition if the similarity between the text and the cluster is higher than a threshold, otherwise it is assumed that the text's characteristics are already conveyed in *Cm*. When the similarity between the text and *Cm* is lower than the pre-defined threshold, the text is assigned to a new cluster. Two similarity metrics were considered by the model: the Dice coefficient, and an adaptation of Dice in which common terms contribute with different weights according to their POS tag. Although *ArteCM* has the ability to create new clusters adaptively, it does not provide any pruning or merging strategies to help reducing the computational complexity of the algorithm. Thus, clusters are only locally modified. Experimental evaluation was based on three review datasets with varying topics (PDAs, cell phones and printers). A standard SinglePass implementation and a k-Means with randomly selected cluster seeds were selected as baseline. *ArteCM* was able to improve the best computational times of both SinglePass and k-Means for the biggest dataset. In conclusion, the authors claimed that *ArteCM* is capable of achieving both high clustering quality and low computational complexity.

Tsur *et al.* (2013) proposed an efficient Multi-stage Clustering algorithm (named SMSC) that divided clustering into two distinctive tasks. First, it performs a batch clustering of a data subset and then an online clustering of a data stream. The batch clustering is divided into three tasks. First, a collection of non-sparse texts is created by aggregating all texts sharing the same tag. The number of non-sparse texts corresponds to the number of tags. If a text contains more than one tag, it is aggregated to all of the corresponding non-sparse texts. Second, the content of the non-sparse texts is clustered using k-Means. Third, all individual texts are clustered based on the label of their corresponding non-sparse text. As the described stages are supposed to be efficient, they could be regularly repeated in order to increase accuracy and add new hashtags or trends. Once the batch clustering is finished, the centroids of each cluster are computed and the online stage starts. Each arriving text is compared to the k centroids and assigned to the closest cluster. In the case the arriving text contains a tag, it is assigned to the cluster corresponding to such tag. It is worth noting that once the batch clustering is finished, the online clustering could be performed in linear time. The selected batch clustering algorithm could be replaced or adapted to different clustering needs. Experimental evaluation was based on one million tweets collected between June and December 2009. Three clustering algorithms were selected as baselines for comparison: a distribution-based clustering (it assigns tweets to clusters according to the manually defined class distribution and, in consequence, cluster sizes correspond to class sizes), standard k-Means and a modified k-Means designed for Web scale sparse data (WSFkM) (Sculley, 2010). The batch clustering of 1 000 manually selected tweets showed similar results for k-Means and the distribution-based clustering, reinforcing the idea that clustering sparse data is difficult, specifically when cluster definitions are fuzzy. SMSC outperformed the distributed-based clustering for all metrics, and traditional k-Means for all but one metric. Although WSFkM was designed to address large and sparse collections of Web pages, it performed poorly when clustering tweets. The authors found that WSFkM only created a cluster comprising the biggest proportion of tweets and the remaining clusters only comprised one or two tweets. Although k-Means and the online component of SMSC obtained similar performance results, k-Means is more time consuming as it iterates until a stable partitioning is found, which makes it unsuitable for massive stream clustering, whereas SMSC can cluster tweets in a linear time. The authors concluded that even when related hashtags tend to appear together, their co-occurrence is still relatively sparse when

https://doi.org/10.1017/S0269888919000018

compared to the aggregated texts, reducing the amount of available contextual information and, thus, decreasing clustering performance.

Popovici *et al.* (2014) extended the *DenStream* and *DBSCAN* algorithms. The technique aims at addressing the requirements of data stream mining such as fast incremental processing of new stream texts, compactness of data representation and efficient identification of changes in the evolving clustering models. Alike the *DenStream* algorithm, a set of core and outlier micro-clusters is incrementally maintained. Clusters might change their role due to outdated micro-clusters fading into outliers and new micro-clusters being created. An outlier buffer is added to separate both types of clusters to speed-up the process. A lightweight variant of the general macro-clustering approach *DBSCAN* is also applied on the micro-clusters as virtual points. The authors introduced the concept of sub-clusters within micro-clusters that are incrementally maintained to supervise the appearance of interesting sub-clusters as part of a major cluster. Cluster's centroids are efficiently and accurately estimated by defining statistical features (common and proper nouns) extracted from each cluster by means of POS tagging techniques. Incoming texts are assigned to the closest cluster. Then, they are reassigned to the closest sub-cluster inside the micro-cluster by comparing them to the sub-clusters summaries. The sub-cluster summaries comprise the number of contained data points, the linear sums of each feature, the linear sums of the co-occurrences per feature and the linear sums of occurrences of a feature in a particular window. To increase efficiency, similarity and summary values are additively updated. In all cases, the similarity between instances and clusters is computed by means of the Cosine similarity. Experimental evaluation was performed in the context of the SNOW 2014 Data Challenge and based on more than a million tweets collected on February 2014 belonging to 70 topics. The approach obtained competitive diversity, readability and coherence results as compared to the other participant approaches. Nonetheless, although the approach seemed promising, its performance was not sufficient to win the data challenge nor outperform simpler approaches based on aggressive feature selection and simpler clustering techniques.

4.3.4 Density-based techniques

Density-based techniques cluster nodes according to edge density characteristics. They can discover clusters of arbitrary shapes and sizes by describing a simple equation based on the overall density function. Generally, these techniques require the definition of several parameters. In turn, choosing the wrong parameter setting could significantly affect their performance. Miller *et al.* (2013) proposed two online clustering algorithms, *DenStream* and *StreamKM++*, for spam detection in *Twitter*. *DenStream* is a density clustering algorithm designed for stream environments. Unlike the traditional DBSCAN, *DenStream* clusters instances in core-micro-clusters comprising three attributes: weight, centre and radius. Initially, the algorithm receives a set of texts to be clustered and creates the initial core-micro-clusters. When new texts are received, the distance to the nearest core-cluster is computed. If the new instance falls within the radius of the nearest core-cluster, it is merged and cluster attributes are updated. Otherwise, the new instance is assigned to a new cluster. On the other hand, *StreamKM++* builds the clusters based on weighted instances (the coresets), which proved to be a good approximation to the real cluster structure. Unlike k-Means++ (Arthur & Vassilvitskii, 2007), *StreamKM++* has the ability to update the clusters as new instances arrive. To begin clustering, a small set of texts is selected to build the coresets. k-Means++ clustering is performed several times to select the tightest clustering model. When new instances arrive, the distance between the new instance and the nearest cluster is computed. After a pre-defined number of texts arrives in the stream, the clustering model is re-built. Experimental evaluation was based on 3, 239 tweets posted by different users. Along with the content information provided by each tweet, information regarding the user who posted it was also considered (e.g., follower, friend, favourites, tweet and re-tweet count, age and follower ratio). StreamKM++ was observed to be better suited than *DenStream* for detecting all spam instances regardless of incorrectly classifying normal ones. The combination of both algorithms outperformed the results of each individual algorithm, improving the false positives rate obtained by StreamKM++ and detecting all the spammers. Finally, regarding the set of selected features, accuracy and false positive rate results were improved when the content of each tweet was used during clustering for both algorithms, reinforcing the importance of textual features for spam detection.

As previously mentioned, event detection is an important task in social media. However, the short-texts reporting events are usually overwhelmed by a high number of unrelated texts that pollute the stream, thus requiring flexible and scalable techniques. Moreover, traditional event detection techniques usually require to determine the number of events that should be detected, which is not possible in social media, given its real time nature. In this regard, Weng and Lee (2011) presented the EDCoW technique that aims at tackling the mentioned problems. The underlying assumption of the technique is that some related words would show an increase in the usage when an event is happening. Hence, an event can be conventionally represented by a number of words showing burst in appearance count. EDCoW comprises four steps. First, it builds signals for individual words that capture only the bursts in words' appearance. These signals can be efficiently computed by wavelet analysis. Second, it filters the trivial words by looking at their corresponding signal auto-correlations. Third, it measures the cross-correlation between signals. Four, it detects the events by clustering the signals together using a modularity-based graph clustering, solved with a scalable eigenvalue algorithm. It is worth noting that EDCoW does not require to specify the number of events to detect, as it will automatically generate different number of events based on the texts' discussions. To differentiate the events from trivial ones, EDCoW also quantifies the events' significance based on two factors: the number of words, and the cross-correlation amongst the words related to the event. Experimental evaluation was based on a manually tagged Singapore-based *Twitter* dataset. EDCoW was only compared with LDA, as the authors considered that event detection can be considered equivalent to topic modelling. For the LDA evaluation, all tweets published on the same day were aggregated into a single document, and then topics were detected over the document collection. The reporting of results instead of focusing on precision and recall focused on the interpretability of clusters. According to the authors, EDCoW discovered more easily interpretable clusters. Although the approach seems promising, more evaluations are needed, particularly comparisons to other state-of-the-art techniques. Moreover, as the authors proposed this approach in the context of real time social media, a computational complexity analysis is needed to effectively assess the capabilities of the approach.

In addition to the complexity added by short-texts' sparsity, the massive use of social media generates an extensive amount of data that poses new challenges to machine learning techniques. Most techniques assume that the available RAM memory can fit all data simultaneously. However, the enlargement of RAM memory capacity cannot cope with the exponential growing of data, making the initial assumption invalid. Additionally, algorithms decrease their performance when executing on small-size RAM memory settings. For example, the Louvain algorithm (Blondel *et al.*, 2008) requires accessing to the entire dataset sequentially and frequently, and as a result, when executing on small-size RAM settings it requires more page swapping, which in turn decreases its performance. In this context, Kim *et al.* (2013) proposed an adaptation of the Louvain algorithm aiming at obtaining good performance even with huge datasets. The adaptation partitions the data to fit the available memory, and then processes each partition individually. As each partition can be completely loaded into memory, the algorithm does not require additional data swapping. Experimental evaluation was based on two *Twitter* datasets with at least 2.6 million nodes and 10 million topological connections between them. Although results seemed promising regarding the time speed-up and the increase in the data volume that can be analyzed, the partitioned algorithm could not guarantee the same modularity quality as the original algorithm. Given that the adapted algorithm performs a random partition of data, which is unrelated to the underlying cluster structure, it is unable to reproduce the clustering results of the original algorithm. According to the authors, performing a post-processing step could improve the quality of results, although this remains to be experimentally proved.

4.4 Summary

Text clustering is a fundamental problem in text mining and information retrieval, aiming at grouping similar texts together such that clusters exhibit higher intra-cluster similarity than inter-cluster similarity. Note that no knowledge regarding the existence of classes is needed in advanced. As with the classification task, short-texts sparseness and high volume pose several challenges to clustering techniques. Table 2 summarises the main characteristics of the reviewed clustering techniques, based on the type

Table 2 Summary of clustering techniques

	Task	Critical time	Binary or multi-class?	Updates over time?	Computational complexity	Type of features	Includes pre-processing?	Evaluation
Zhang *et al.* (2014)	Text clustering	Online	Multi-cluster	Yes	Low	Re-posting behaviour of users	No	Small dataset extracted from Sina Weibo
Weibo Tu and Ding (2012)	Semi-supervised event detection	Online	Multi-cluster	No	Low	Textual	TF-IDF weighting, nouns and verbs carried more weight	10 590 short-texts from Sina Weibo
Wang *et al.* (2016b)	Semi-supervised text clustering	Batch	Multi-cluster	Yes	Medium to high	Textual	TF-IDF weighting	Four datasets comprising questions and short-texts
Li *et al.* (2012)	Event detection	Batch	Multi-cluster	No	Medium	Textual	Segmentation	Tweets published by Singapore-based users
Ferrara *et al.* (2013)	Topic extraction	Batch	Multi-cluster	No	Medium	Textual	No	Medium-scale number of tweets regarding the US primaries in 2012
Parikh and Karlapalem (2013)	Event detection	Batch	Multi-cluster	No	Medium	Textual	Tokenisation, stopword removal and stemming	Large-scale tweet dataset related to the broadcasting domain

Stilo and Velardi (2017)	Hashtag clustering	Online	Multi-cluster	No	Low/medium	Textual	Unknown	5.1 million multilingual tweets
Yin *et al.* (2018)	Text clustering	Online	Multi-cluster	Yes	Low/medium	Textual	Lowercasing, stopword removal and stemming	Tweets, news and a combination of them
Liang *et al.* (2016)	Topic extraction	Online	Multi-cluster	Yes	Medium	Textual	No	Over 369 millions of English tweets
Jia *et al.* (2018)	Concept extraction	Batch	Multi-cluster	No	Medium	Textual	Stemming, removal of duplicates, digits, stopwords and words with frequency lower than threshold	Two Chinese and three English datasets, including questions and tweets
Kim *et al.* (2012)	Topic extraction	Batch	Multi-cluster	No	Medium	Textual	Only proper nouns, word sequences beginning with capital letters and phrases enclosed by quotation marks were kept	Tweets related to four popular TV shows
Ni *et al.* (2011)	Topic extraction	Batch	Multi-cluster	No	Medium	Textual	Segmentation and stopword removal	Manually classified questions from *BuyAns* and search snippets
Yang and Ng (2009)	Web opinion clustering	Online	Multi-cluster	Yes	Medium	Textual	Only the twenty terms with the highest TF-IDF scores are selected.	Over than 400 threads of Politics extracted from *MySpace*.

Table 2 Continued.

	Task	Critical time	Binary or multi-class?	Updates over time?	Computational complexity	Type of features	Includes pre-processing?	Evaluation
Carullo *et al.* (2009)	Text clustering	Online	Multi-cluster	Yes	Low	Textual	Unknown	Three datasets of short descriptions of Web products
Tsur *et al.* (2013)	Topic detection	Batch and online	Multi-cluster	No	Low	Textual and statistical	Unknown	Over than 417 000 000 tweets
Popovici *et al.* (2014)	Spam detection	Online	Binary	Yes	Medium.	Textual	Unknown	Over than 3 000 *Twitter* accounts, each with a sample tweet.
Miller *et al.* (2013)	Spam detection	Online	Binary	Yes	Medium	Textual and user-based	Unknown	Over than 3 000 *Twitter* accounts, each with a sample tweet
Weng and Lee (2011)	Event detection	Batch	Multi-cluster	No	Medium	Textual	Words appearing less than five times and with patterns repeating more than two times were removed	Tweets published by Singapore-based users
Kim *et al.* (2013)	Community and topic detection	Batch	Multi-cluster	No	Medium	Social and textual	No	Over than 41 million *Twitter* accounts and their corresponding tweets

of features used, whether they are presented for a batch or online environments, whether clusters are updated over time and how evaluation was performed, amongst other relevant characteristics. Note that most of the review techniques neglect the existence of information sources other than the textual content. Additionally, in most cases, authors did not perform a statistical analysis of the significance of differences, which hinders the comparability of results and the real assessment of the performance of the review approaches. Finally, it is worth noting that the authors neglected mentioning the capabilities of techniques for finding clusters of irregular shapes, handling outliers or whether the techniques are sensitive to the order in which instances are processed. Nonetheless, the review shows that the designed clustering techniques are more prepared than the reviewed classification techniques for tackling the challenges posed by text learning in social media.

5 Discussion

Although text learning has received considerable attention during the last decades, most studies did not focus on facing the challenges posed by the nature of short-texts and especially social media data. At present day, efficient and scalable learning is an important requirement in numerous large-scale social applications. In this context, the main shortcomings and challenges of the reviewed techniques can be summarised as follows.

Curse of dimensionality. As previously mentioned, text mining and learning tasks are characterised by the high dimensionality of their feature space where most terms have a low frequency. Furthermore, as data dimensionality increases, the volume of the feature space increases rapidly, so that the available data become sparse. Additionally, the linked nature of social media data generates new information, such as who creates the posts (post authorship, i.e., user–post relations), and who is friend of whom (friendship, i.e., user–user relations), which can be added to the feature space (Tang & Liu, 2012). As the feature space grows, the number of features available for the model also increases, thus increasing the computational complexity of techniques. In the case of clustering, as the dimensionality of the dataset increases, distance metrics might become meaningless. Moreover, the continuous growth of social media data puts in jeopardy the scalability of current techniques, especially in real-time tasks. For example, most techniques require all data to be loaded into memory, which might not be possible when analyzing social media data.

In response to the mentioned challenges, feature selection techniques should be applied and new learning approaches specifically designed for short-texts should be developed. In this regard, several classification (Yuan *et al.*, 2012; Collins *et al.*, 2015; Cui *et al.*, 2016) and clustering (Yang & Ng, 2009; Ni *et al.*, 2011; Weng & Lee, 2011; Tu & Ding, 2012; Li *et al.*, 2012; Kim *et al.*, 2012; Parikh & Karlapalem, 2013; Wang *et al.*, 2016b; Jia *et al.*, 2018; Yin *et al.*, 2018) techniques included pre-processing steps aiming at reducing the dimensionality of the feature space. In all cases, the applied pre-processing included traditional techniques applied to long-texts. Interestingly, none of techniques evaluated on social media data included lexical normalisation, tokenising nor POS taggers designed for social media texts. Conversely, the other reviewed techniques did either not apply or not disclose the use of any feature selection nor pre-processing techniques. These last techniques presented a low to medium computational complexity. Thereby, it could be inferred that the technique is not sensitive to the high-dimensional feature space, or that evaluations did not specifically test the scalability of the presented approaches.

Temporal relevance of Data. In dynamically changing environments, such as social networking sites, data distribution can change over time, yielding the phenomenon of ‘concept drift’ (Gama *et al.*, 2014). Social media can be seen as a particular form of a temporal data stream in which concepts appear and disappear often, as users post quickly as an event occurs (e.g., natural disasters), and then naturally disappear after a short period (e.g., a few days), reoccurring or not some time later. Concept drift refers to changes in the conditional distribution of the output, while the input distribution stays unchanged. Learning algorithms operating in these settings need mechanisms to detect and adapt to the evolution of data over time, otherwise their performance will degrade. Hidden changes in context can also cause a change of the underlying data distribution (Tsymbal, 2004), which might often lead to the need of updating the current model as the model's error might not be longer acceptable with the new data distribution. Ideally, techniques handling concept drift should be capable of quickly adapting to changes, be robust to noise to differentiate between noise and real concept drift, and recognise and treat recurring contexts. This is especially important in real-time applications.

Some of the reviewed classification (Sedhai & Sun, 2015; Li *et al.*, 2017, 2018) and clustering (Yang & Ng, 2009; Carullo *et al.*, 2009; Miller *et al.*, 2013; Zhang *et al.*, 2014; Popovici *et al.*, 2014; Liang *et al.*, 2016; Wang *et al.*, 2016b; Yin *et al.*, 2018) techniques proposed to update the learned model as new instances are known, as a means to keep up with the new trends that might appear. In the other cases, the authors did not propose any mechanism to update the models. Nonetheless, the techniques could still be applied in dynamic environments, provided the training process is redone after a certain batch of instances is known. Given that trends do not only appear, but can also disappear, it might be important to not only add new instances, but also discard old ones. In this regard, only one of the reviewed techniques (Yin *et al.*, 2018) studied the effect of removing old instances, and concluded that it allowed to improve not only the quality of clusters, but also the spacial and computational efficiency of the technique.

Availability of labelled examples. Most techniques assume the existence of a fixed set of instances. However, in real-world applications, training instances might not be available in advance, as they could arrive in a sequential stream, or it could be difficult to collect a full training set (Wang *et al.*, 2014). In the case of classification, the problem aggravates as the number of distinct classes increases, and therefore the size of the needed training set also increases (Forman, 2004). Hence, some classes will inevitably be more difficult than others to predict. Forman (2004) hypothesised that, in the case it is difficult to get good predictive features for some classes, techniques will focus on predictors for other easier classes, thus ignoring the difficult ones. This problem worsens in a real-time scenario. On the other hand, given that clustering techniques do not need such labelled data, they have become useful for real-time analysis.

As regards the reviewed classification techniques, only a few of them (Romero *et al.*, 2013; Sedhai & Sun, 2015; Li *et al.*, 2017) required few or none labelled training data. Particularly, Romero *et al.* (2013) relied on describing classes by means of ontologies. Although the ontology alleviates the need of labelled instances, it still requires to have domain knowledge to determine the number of classes and create the class representations. In this regard, this technique could be affected by the concept drift problems. On the other hand, the technique presented by (Li *et al.*, 2017) only required a small labelled training set that was later enriched with unlabelled instances, which allowed it to better cope with the concept drift problems. The problem worsens in online applications in which new classes might also appear. On the other hand, although clustering techniques do not need labelled instances, they might require the definition or estimation of the number of clusters to discover for achieving a good partitioning of instances, which is also difficult in online scenarios.

Streaming and online learning. One of the difficulties posed by social media data is the massive volume of data arriving in streams to analyze. In this regard, one of the current challenges is to find adequate approaches that allow the analysis of such streams (Gandomi & Haider, 2015). Although sampling and dimensionality reduction methods can improve the scalability and speed of algorithms, the data growth is bigger and faster than the memory and processor advancements. As a result, single machine (either single or multi-thread) algorithms might not be able to handle the increasing amount of data, highlighting the need for distributed and parallel algorithms. During the last few years, a new paradigm has arisen (Witten *et al.*, 2016) in which algorithms should be developed to naturally cope with datasets that are bigger than the memory size, and perhaps indefinitely large. This paradigm assumes that each instance can be inspected only once, and must then be discarded. It is worth noting that the learning algorithm has no control over the order in which instances are processed, and should update the trained model incrementally as new instances arrive. Typically, these algorithms operate indefinitely, whilst using limited memory. Although, as time advances, the model might grow, it should not be allowed to grow boundless. In the case of real-time applications, algorithms should be able to process instances faster than they arrive, and preferably in a fixed, constant and small time bound.

https://doi.org/10.1017/S0269888919000018

Two stream learning scenarios are possible, depending on whether there are sufficient instances for learning a model (Aggarwal, 2014). In the first case, instances might be available for batch learning, and then new instances arrive in the form of a stream. This implies that only the classification or clustering of instances needs to be efficient. As a result, any of the reviewed techniques could be applied to this scenario. In the second case, as instances are continuously arriving, the patterns in the training instances might change over time (as explained for the concept drift), thus requiring techniques that could update the learned models to the new patterns. As noted, considering the reviewed techniques, a higher proportion of clustering (Yang & Ng, 2009; Carullo *et al.*, 2009; Tu & Ding, 2012; Tsur *et al.*, 2013; Miller *et al.*, 2013; Zhang *et al.*, 2014; Popovici *et al.*, 2014; Liang *et al.*, 2016; Stilo & Velardi, 2017; Yin *et al.*, 2018) than classification (Sedhai & Sun, 2015; Li *et al.*, 2018; Ravi & Kozareva, 2018) techniques were suitable for online learning, even including updating mechanisms. Interestingly, batch classification techniques did not yield a high computational complexity, meaning that, although is not the ideal, those techniques could be also used for classifying instances in real-time at the cost of retraining the whole model at certain intervals of time, simulating a batch-based incremental learning approach. Additionally, decay factors should be introduced to guarantee that the new instances are weighted more significantly in the learned model, thus coping with the concept drift problem. Classification over streams could also be affected by the distribution of classes in the stream, as some of the classes might not frequently arrive. In these cases, classification becomes extremely challenging as often it might be more important to detect the infrequent classes than the frequent ones.

Heterogeneous data. Most of the reviewed techniques assume that the type of objects or links is unique and homogeneous, for example, an author collaboration network and a friendship network (Shi *et al.*, 2017). These homogeneous networks are usually extracted from real systems by naïvely ignoring the heterogeneity of data, or by restricting the relations amongst instances to only one type. Nonetheless, social media data offer multiple and diverse information sources, hence features might not only be limited to text. For instance, as previously stated, the linked nature of social media data generates new information that can be added to the feature space (Tang & Liu, 2012). Moreover, new information sources, and thus features, can be inferred from the explicit relations between instances. It is worth noting that each information source could include errors and missing values. In this regard, combining multiple information sources could compensate for incomplete information or noise, allowing to validate trustworthy relationships and train more reliable and accurate models. Nonetheless, this poses new challenges. For example, which information sources to combine. It is worth noting that choosing the information sources to integrate depends on the elements available on the social network under analysis, such as the characteristics and semantic of social relations, the semantics of the messages that users' exchange, or the content of such messages, amongst others. Second, how to integrate the selected information sources by assessing their importance in the context of the different social media sites.

Although most of the reviewed techniques were designed for short-texts extracted from social media, the majority of them only considered textual features. As an exception (Tsur *et al.*, 2013; Miller *et al.*, 2013; Kim *et al.*, 2013; Zhang *et al.*, 2014; Sedhai & Sun, 2015) included additional information sources in the classification or clustering process, such as behavioural aspects related to social ties and relations (Kim *et al.*, 2013) or profile and statistical features (Miller *et al.*, 2013; Tsur *et al.*, 2013; Zhang *et al.*, 2014; Sedhai & Sun, 2015). Those techniques that only leveraged on textual features can be naïvely enriched with information belonging to other information sources by simulating new categorical features. Nonetheless, such adaptation might not convey the semantics of each particular source, which could hinder the learning process.

6 Conclusions

Text mining refers to a knowledge discovery process aiming at the extraction of interesting and non-trivial patterns from natural language, which is characterised by the high dimensionality of its feature space. With the advent of short-texts, new challenges in automatic learning processes are faced. First, unlike traditional and long-texts, in the context of social media, texts are usually noisier, less topic-focused and

https://doi.org/10.1017/S0269888919000018

shorter. Given such brevity, traditional similarity or overlapping-based approaches might not achieve high performance. Second, the linked nature of social media data generates new information sources (e.g., who creates the posts, and who is friend of whom), which although they can be added to the feature space, they are neglected by most approaches. Third, the continuous growth of social media data puts in jeopardy the scalability of current techniques, for example due to the necessity of loading all data into memory. In response to these challenges, new learning approaches specifically designed for short-texts have arisen. This survey reviews the field of short-text learning, focusing on classification and clustering techniques, as they are two of the most frequent learning tasks. The survey illustrates how the classification and clustering of social media short-texts have been tackled discussing limitations and current unsolved issues of the existing techniques.

As the analysis showed, several of the reviewed techniques might be suitable for performing text learning in online or streaming environments, as social media applications require. However, certain considerations must be taken into account. First, the selected technique needs to be efficient in its space management. For example, when considering social media data, it might not be possible to fit all data into memory. Thus, techniques should be able to efficiently analyze data instances. Second, it should have a low computational complexity. Particularly, for classification techniques to achieve real-time updates of the model, the training complexity should be low. Third, the technique should be able to cope with dynamic environments in which the learning model has to be periodically updated. Fourth, in the case of short-text classification, as a great number of labelled texts might not be available, the technique should not only depend on labelled instances for training. Techniques requiring few or no training instances should be preferable. Fifth, it would be also of great importance to consider parallelisable and incremental techniques. Similarly, model updates should be preferably made in background. Sixth, techniques should not be restricted to binary categorisation settings. In the context of social media data, new topics and categories constantly emerge. Consequently, techniques should be able to adapt to new appearing categories or topics. Finally, in the case of clustering, techniques requiring to know in advance the number of clusters should be avoided, as it might be difficult define it.

References

Aggarwal, C. C. 2014. A survey of stream classification algorithms. In *Data Classification: Algorithms and Applications*, Aggarwal, C. C. (ed). CRC Press, 245–274.

Aggarwal, C. C. & Zhai, C. X. 2012. A survey of text classification algorithms. In *Mining Text Data*, Aggarwal, C. C. & Zhai, C. X. (eds). Springer US, 163–222. ISBN 978-1-4614-3222-7.

Alelyani, S., Tang, J. & Liu, H. 2013. Feature selection for clustering: a review. In *Data Clustering: Algorithms and Applications*. Chapman and Hall/CRC, 29–60.

Arthur, D. & Vassilvitskii, S. 2007. k-means++: the advantages of careful seeding. In *SODA '07: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1027–1035. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA. ISBN 978-0-898716-24-5.

Asur, S. & Huberman, B. A. 2010. Predicting the future with social media. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, **1**, 492–499.

Becker, H., Naaman, M. & Gravano, L. 2011. Beyond trending topics: real-world event identification on Twitter. In *Fifth International AAAI Conference on Weblogs and Social Media*.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, **2008**(10), P10008.

Broder, A. Z. 1997. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, 21–29.

Carullo, M., Binaghi, E. & Gallo, I. 2009. An online document clustering technique for short web contents. *Pattern Recognition Letters* **30**(10), 870–876.

Ciampaglia, G. L., Shiralkar, P., Rocha, L. M., Bollen, J., Menczer, F. & Flammini, A. 2015. Computational fact checking from knowledge networks. *PLOS ONE* **10**(6), 1–13 .

Collins, R., May, D., Weinthal, N. & Wicentowski, R. 2015. SWAT-CMW: classification of Twitter emotional polarity using a multiple-classifier decision schema and enhanced emotion tagging. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 669–672. Association for Computational Linguistics.

Croft, W. B., Metzler, D. & Strohman, T. 2010. *Search Engines: Information Retrieval in Practice*, **283**. Addison-Wesley Reading.

Cui, R., Agrawal, G., Ramnath, R. & Khuc, V. 2016. Ensemble of heterogeneous classifiers for improving automated tweet classification. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, 1045–1052.

Dai, X., Bikdash, M. & Meyer, B. 2017. From social media to public health surveillance: word embedding based clustering method for Twitter classification. In *SoutheastCon 2017*, 1–7.

de la Rosa, G. R., Montes-y-Gémez, M. Solorio, T. & Pineda, L. V. 2013. A document is known by the company it keeps: neighborhood consensus for short text categorization. *Language Resources and Evaluation* **47**(1), 127–149.

Deutsch, P. 1996. DEFLATE Compressed Data Format Specification version 1.3. RFC 1951 (Informational).

Dietterich, T. G. 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems*. Springer, 1–15. ISBN 978-3-540-45014-6.

Efron, M., Lin, J., He, J. & de Vries, A. 2014. Temporal feedback for tweet search with non-parametric density estimation. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '14*, 33–42. ACM. ISBN 978-1-4503-2257-7.

Ferrara, E., JafariAsbagh, M., Varol, O., Qazvinian, V., Menczer, F. & Flammini, A. 2013. Clustering memes in social media. In *Proceedings of IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*.

Forman, G. 2004. A pitfall and solution in multi-class feature selection for text classification. In *ICML*, Brodley, C. E. (ed), ACM International Conference Proceeding Series, **69**. ACM.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M. & Bouchachia, A. 2014. A survey on concept drift adaptation. *ACM Computing Surveys* **46**(4), 44:1–44:37. ISSN 0360-0300.

Gandomi, A. & Haider, M. 2015. Beyond the hype: big data concepts, methods, and analytics. *International Journal of Information Management* **35**(2), 137–144. ISSN 0268-4012.

Guyon, I. & Elisseeff, A. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research* **3**, 1157–1182.

Hu, M. & Liu, B. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04*, 168–177. ACM. ISBN 1-58113-888-1.

Irfan, R., King, C. K., Grages, D., Ewen, S., Khan, S. U., Madani, S. A., Kolodziej, J., Wang, L., Chen, D. & Rayes, A. 2015. A survey on text mining in social networks. *The Knowledge Engineering Review* **30**(2), 157–170.

Iwata, T., Watanabe, S., Yamada, T. & Ueda, N. 2009. Topic tracking model for analyzing consumer purchase behavior. In *IJCAI*, **9**, 1427–1432.

Jain, A. K. & Dubes, R. C. 1988. *Algorithms for Clustering Data*. Prentice-Hall, Inc. ISBN 0-13-022278-X.

Jia, C., Carson, M. B., Wang, X. & Yu, J. 2018. Concept decompositions for short text clustering by identifying word communities. *Pattern Recognition* **76**, 691–703. ISSN 0031-3203.

Kang, J. H., Lerman, K. & Plangprasopchok, A. 2010. Analyzing microblogs with affinity propagation. In *Proceedings of the First Workshop on Social Media Analytics*, 67–70. ACM.

Khan, F. H., Bashir, S. & Qamar, U. 2014. Tom: Twitter opinion mining framework using hybrid classification scheme. *Decision Support Systems* **57**, 245–257. ISSN 0167-9236.

Kim, K., Chung, B.-S., Choi, Y., Lee, S., Jung, J.-Y. & Park, J. 2014. Language independent semantic kernels for short-text classification. *Expert Systems with Applications* **41**(2), 735–743. ISSN 0957-4174.

Kim, S., Jeon, S., Kim, J., Park, Y.-H. & Yu, H. 2012. Finding core topics: topic extraction with clustering on tweet. In *2012 Second International Conference on Cloud and Green Computing (CGC)*, 777–782.

Kim, Y.-H., Seo, S., Ha, Y.-H., Lim, S. & Yoon, Y. 2013. Two applications of clustering techniques to Twitter: community detection and issue extraction. *Discrete Dynamics in Nature and Society* **2013**.

Li, C., Sun, A. & Datta, A. 2012. Twevent: segment-based event detection from tweets. In *CIKM*, Chen, X. W., Lebanon, G., Wang, H. & Zaki, M. J. (eds). ACM, 155–164. ISBN 978-1-4503-1156-4.

Li, J., Khan, S. U., Li, Q., Ghani, N., Min-Allah, N., Bouvry, P. & Zhang, W. 2011a. Efficient data sharing over large-scale distributed communities. In *Intelligent Decision Systems in Large-Scale Distributed Environments*. Springer, 149–164.

Li, J., Li, Q., Khan, S. U. & Ghani, N. 2011b. Community-based cloud for emergency management. In *2011 6th International Conference on System of Systems Engineering*, 55–60.

Li, P., He, L., Wang, H., Hu, X., Zhang, Y., Li, L. & Wu, X. 2018. Learning from short text streams with topic drifts. *IEEE Transactions on Cybernetics* **48**(9), 2697–2711. ISSN 2168-2267. doi: 10.1109/TCYB.2017.2748598.

Li, S., Wang, Z., Zhou, G. & Lee, S. Y. M. 2011c. Semi-supervised learning for imbalanced sentiment classification. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume 3, IJCAI '11*, 1826–1831. AAAI Press. ISBN 978-1-57735-515-1.

Li, X., Yan, L., Qin, N. & Ran, H. 2017. A novel semi-supervised short text classification algorithm based on fusion similarity. In *Intelligent Computing Methodologies*, Huang, D.-S., Hussain, A., Han, K. & Gromiha, M. M. (eds). Springer International Publishing, 309–319. ISBN 978-3-319-63315-2.

Liang, S., Yilmaz, E. & Kanoulas, E. 2016. Dynamic clustering of streaming short documents. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, 995–1004. ACM, New York, NY, USA. ISBN 978-1-4503-4232-2.

Lifna, C. S. & Vijayalakshmi, M. 2015. Identifying concept-drift in Twitter streams. *Procedia Computer Science* **45**, 86–94. ISSN 1877-0509. International Conference on Advanced Computing Technologies and Applications (ICACTA).

Lin, J., Keogh, E., Lonardi, S. & Chiu, B. 2003. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, DMKD '03*, 2–11. ACM.

Liu, H. & Yu, L. 2005. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering* **17**(4), 491–502.

Losing, V., Hammer, B. & Wersing, H. 2018. Incremental on-line learning: a review and comparison of state of the art algorithms. *Neurocomputing* **275**,1261–1274. ISSN 0925-2312.

Mathew, K. & Issac, B. 2011. Intelligent spam classification for mobile text message. In *2011 International Conference on Computer Science and Network Technology (ICCSNT)*, **1**, 101–105.

Miller, Z., Dickinson, B., Deitrick, W., Hu, W. & Wang, A. H. 2013. Twitter spammer detection using data stream clustering. *Information Sciences* **260**, 64–73. ISSN 0020-0255.

Nakov, P., Rosenthal, S., Kiritchenko, S., Mohammad, S. M., Kozareva, Z. Ritter, A., Stoyanov, V. & Zhu, X. 2016. Developing a successful SemEval task in sentiment analysis of Twitter and other social media texts. *Language Resources and Evaluation* **50**(1), 35–65. ISSN 1574-020X.

Ni, X., Quan, X., Lu, Z., Wenyin, L. & Hua, B. 2011. Short text clustering by finding core terms. *Knowledge and Information Systems* **27**(3), 345–365.

Nishida, K., Banno, R., Fujimura, K. & Hoshide, T. 2011. Tweet classification by data compression. In *Proceedings of the 2011 International Workshop on DETecting and Exploiting Cultural DiversiTy on the Social Web, DETECT '11*, 29–34. ACM. ISBN 978-1-4503-0962-2.

Oh, O., Agrawal, M. & Rao, H. R. 2011. Information control and terrorism: tracking the Mumbai terrorist attack through Twitter. *Information Systems Frontiers* **13**(1), 33–43. ISSN 1572-9419.

Parikh, R. & Karlapalem, K. 2013. ET: events from tweets. In *WWW (Companion Volume)*, Carr, L., Laender, A. H. F., Lóscio, B. F. King, I., Fontoura, M., Vrandecic, D., Aroyo, L., de Oliveira, J. P. M., Lima, F. & Wilde, E. (eds), 613–620. International World Wide Web Conferences Steering Committee/ACM. ISBN 978-1-4503-2038-2.

Phan, X.-H., Nguyen, L.-M. & Horiguchi, S. 2008. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *WWW '08: Proceeding of the 17th International Conference on World Wide Web*, 91–100. ACM. ISBN 978-1-60558-085-2.

Popovici, R., Weiler, A. & Grossniklaus, M. 2014. On-line clustering for real-time topic detection in social media streaming data. In *SNOW-DC@ WWW*, 57–63.

Prusa, J., Khoshgoftaar, T. M. & Dittman, D. J. 2015. Using ensemble learners to improve classifier performance on tweet sentiment data. In *2015 IEEE International Conference on Information Reuse and Integration*, 252–257.

Prusa, J. D., Khoshgoftaar, T. M. & Seliya, N. 2016. Enhancing ensemble learners with data sampling on high-dimensional imbalanced tweet sentiment data. In *FLAIRS Conference*, 322–328.

Rangrej, A., Kulkarni, S. & Tendulkar, A. V. 2011. Comparative study of clustering techniques for short text documents. In *Proceedings of the 20th International Conference Companion on World Wide Web, WWW '11*, 111–112. ACM. ISBN 978-1-4503-0637-9.

Ravi, S. & Kozareva, Z. 2018. Self-governing neural networks for on-device short text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 804–810. Association for Computational Linguistics.

Romero, F. P., Julián-Iranzo, P., Soto, A., Ferreira-Satler, M. & Gallardo-Casero, J. 2013. Classifying unlabeled short texts using a fuzzy declarative approach. *Language Resources and Evaluation* **47**(1), 151–178. ISSN 1574-020X.

Rosa, K. D. & Ellen, J. 2009. Text classification methodologies applied to micro-text in military chat. In *International Conference on Machine Learning and Applications, 2009, ICMLA '09*, 710–714.

Rosa, K. D., Shah, R., Lin, B., Gershman, A. & Frederking, R. 2011. Topical clustering of tweets. In *Proceedings of the ACM SIGIR: SWSM*.

Saeys, Y., Inza, I. & Larrañaga, P. 2007. A review of feature selection techniques in bioinformatics. *Bioinformatics* **23**(19), 2507–2517.

Sajnani, H., Javanmardi, S., McDonald, D. W. & Lopes, C. V. 2011. Multi-label classification of short text: a study on Wikipedia barnstars. In *Analyzing Microtext, AAAI Workshops* **WS-11-05**. AAAI.

Sander, J., Ester, M., Kriegel, H.-P. & Xu, X. 1998. Density-based clustering in spatial databases: the algorithm GDBSCAN and its applications. *Data Mining and Knowledge Discovery* **2**(2), 169–194.

Sculley, D. 2010. Web-scale k-means clustering. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, 1177–1178. ACM, New York, NY, USA. ISBN 978-1-60558-799-8.

Sebastiani, F. 2002. Machine learning in automated text categorization. *ACM Computing Surveys* **34**(1), 1–47. ISSN 0360-0300.

Sedhai, S. & Sun, A. 2015. HSpam14: a collection of 14 million tweets for hashtag-oriented spam research. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15*, 223–232. ACM. ISBN 978-1-4503-3621-5.

Sedhai, S. & Sun, A. 2018. Semi-supervised spam detection in Twitter stream. *IEEE Transactions on Computational Social Systems* **5**(1), 169–175.

Shi, C., Li, Y., Zhang, J., Sun, Y. & Yu, P. S. 2017. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering* **29**(1), 17–37. ISSN 1041-4347.

Song, G., Ye, Y., Du, X., Huang, X. & Bie, S. 2014. Short text classification: a survey. *Journal of Multimedia* **9**(5), 635.

Stilo, G. & Velardi, P. 2017. Hashtag sense clustering based on temporal similarity. *Computational Linguistics*, **43**(1), 181–200. ISSN 0891-2017.

Su-zhi, Z. & Pei-feng, S. 2011. A new short-text categorization algorithm based on improved KSVM. In *2011 IEEE 3rd International Conference on Communication Software and Networks (ICCSN)*, 154–157.

Tang, J. & Liu, H. 2012. Feature selection with linked data in social media. In *Proceedings of the 12th SIAM International Conference on Data Mining*, 118–128. SIAM/Omnipress. ISBN 978-1-61197-232-0.

Tang, J., Alelyani, S. & Liu, H. 2014. Feature selection for classification: a review. In *Data Classification: Algorithms and Applications*, Aggarwal, C. C. (ed). CRC Press, 37–64. ISBN 978-1-4665-8674-1.

Thelwall, M., Buckley, K. & Paltoglou, G. 2011. Sentiment in Twitter events. *Journal of the American Society for Information Science and Technology* **62**(2), 406–418. ISSN 1532-2890.

Tsur, O., Littman, A. & Rappoport, A. 2013. Efficient clustering of short messages into general domains. Proceedings of the 7th International Conference on Weblogs and Social Media, ICWSM 2013. 621–630.

Tsymbal, A. 2004. The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin* **106**(2).

Tu, H. & Ding, J. 2012. An efficient clustering algorithm for microblogging hot topic detection. In *2012 International Conference on Computer Science Service System (CSSS)*, 738–741.

Wang, J., Zhao, P., Hoi, S. C. H. & Jin, R. 2014. Online feature selection and its applications. *IEEE Transactions on Knowledge and Data Engineering* **26**(3), 698–710. ISSN 1041-4347.

Wang, P., Xu, B., Xu, J., Tian, G., Liu, C.-L. & Hao, H. 2016. Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. *Neurocomputing* **174**, 806–814a. ISSN 0925-2312.

Wang, Z., Mi, H. & Ittycheriah, A. 2016b. Semi-supervised clustering for short text via deep representation learning. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11–12, 2016*, 31–39.

Weller, K., Bruns, A., Burgess, J. & Mahrt, M. 2013. *Twitter and Society*. Peter Lang International Academic Publishers. ISBN 1433121697, 9781433121692.

Weng, J. & Lee, B.-S. 2011. Event detection in Twitter. In *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17–21, 2011*, Adamic, L. A., Baeza-Yates, R. A. & S. Counts (eds). The AAAI Press.

Witten, I. H., Frank, E., Hall, M. A. & Pal, C. J. 2016. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

Wu, W., Li, H., Wang, H. & Zhu, K. Q. 2012. Probase: a probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, SIGMOD '12*, 481–492. ACM. ISBN 978-1-4503-1247-9.

Xu, R. & Wunsch, D. 2005. Survey of clustering algorithms. *IEEE Transactions on Neural Networks* **16**(3), 645–678. ISSN 1045-9227.

Yan, L., Zheng, Y. & Cao., J. 2018. Few-shot learning for short text classification. *Multimedia Tools and Applications* **77**(22), 29799–29810. ISSN 1573-7721.

Yang, C. C. & Ng, T. D. 2009. Web opinions analysis with scalable distance-based clustering. In *ISI*, 65–70. IEEE.

Yang, L., Li, C., Ding, Q. & Li, L. 2013. Combining lexical and semantic features for short text classification. *Procedia Computer Science* **22**:78–86. ISSN 1877-0509. 17th International Conference on Knowledge Based and Intelligent Information and Engineering Systems - KES 2013.

Yin, J. & Wang, J. 2014. A Dirichlet multinomial mixture model-based approach for short text clustering. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*, 233–242. ACM. ISBN 978-1-4503-2956-9.

Yin, J., Chao, D., Liu, Z., Zhang, W., Yu, X. & Wang, J. 2018. Model-based clustering of short text streams. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2634–2642. ACM.

Yu, Y. & Chen, Y. 2012. A novel content based and social network aided online spam short message filter. In *2012 10th World Congress on Intelligent Control and Automation (WCICA)*, 444–449.

Yuan, Q., Cong, G. & Magnenat-Thalmann, N. 2012. Enhancing naive bayes with various smoothing methods for short text classification. In *WWW (Companion Volume)*, Mille, A., Gandon, Misselis, F. L. J., Rabinovich, M. & Staab, S. (eds). ACM, 645–646. ISBN 978-1-4503-1230-1.

Zhai, C. & Lafferty, J. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems* **22**(2), 179–214. ISSN 1046-8188.

Zhang, G., Sun, Y., Xu, M. & Bie, R. 2014. Weibo clustering: a new approach utilizing users' reposting data in social networking services. *Computer Science and Information Systems* **11**(3), 1157–1172.

Zhang, H. & Zhong, G. 2016. Improving short text classification by learning vector representations of both words and hidden topics. *Knowledge-Based Systems* **102**, 76–86. ISSN 0950-7051.

Zubiaga, A., Liakata, M., Procter, R., Bontcheva, K. & Tolmie, P. 2015. Towards detecting rumours in social media. In *AAAI Workshop: AI for Cities*.