Learning From Short Text Streams With Topic Drifts

Article *in* IEEE Transactions on Cybernetics · September 2017

DOI: 10.1109/TCYB.2017.2748598

Peipei Li
Hefei University of Technology

Hu Xuegang
Fujian Medical University

Yuhong Zhang
CHINA INSTITUTE OF INDUSTRIAL RELATIONS

Lei li
Hefei University of Technology

Learning From Short Text Streams With Topic Drifts

Peipei Li, Lu He, Haiyan Wang, Xuegang Hu, Yuhong Zhang, Lei Li, *Senior Member, IEEE*, and Xindong Wu, *Fellow, IEEE*

***Abstract*—Short text streams such as search snippets and micro blogs have been popular on the Web with the emergence of social media. Unlike traditional normal text streams, these data present the characteristics of short length, weak signal, high volume, high velocity, topic drift, etc. Short text stream classification is hence a very challenging and significant task. However, this challenge has received little attention from the research community. Therefore, a new feature extension approach is proposed for short text stream classification with the help of a large-scale semantic network obtained from a Web corpus. It is built on an incremental ensemble classification model for efficiency. First, more semantic contexts based on the senses of terms in short texts are introduced to make up of the data sparsity using the open semantic network, in which all terms are disambiguated by their semantics to reduce the noise impact. Second, a concept cluster-based topic drifting detection method is proposed to effectively track hidden topic drifts. Finally, extensive studies demonstrate that as compared to several well-known concept drifting detection methods in data stream, our approach can detect topic drifts effectively, and it enables handling short text streams effectively while maintaining the efficiency as compared to several state-of-the-art short text classification approaches.**

***Index Terms*—Classification, short text stream, topic drifting.**

I. INTRODUCTION

SHORT texts are prevalent on the Web, no matter in traditional Websites, e.g., news titles and search snippets, or in emerging social media, e.g., micro blogs and tweets. Unlike traditional normal texts such as news articles, short texts refer to the length of the shorter text form. For example, Sina micro-blog and Twitter are new multimedia mini blogs and a Sina micro-blog/tweet has the 140 word/character limit. In recent years, these data swept the world at an alarming rate, and have produced a large quantity of data streams. We call them as short text streams. It is hence challenging for short text stream classification, due to the inherent uniqueness of short text streams such as short length, weak signal and high ambiguity for each short text, and the explosive growth and popularity of short textual content.

Considering the characteristics of short text streams, it is hard to apply the conventional text classification to the model building (such as using bag-of-words [1]) because of the following challenges. First, there are no enough information or statistical signals in short text streams to make the analysis meaningful. Second, it is more difficult to identify the senses of ambiguous words in each short text with limited contexts. Last, existing short text classification methods rarely notice the data stream characteristics in short texts such as high-volume and concept drifts (namely topic drifts in short texts). Thus, it is a challenge in the tackling of short text stream classification due to the efficiency and effectiveness.

To handle of short text classification, existing approaches mainly follow two directions to enrich the short text. The first one is to extend the feature space using the rules or statistical information hidden in the current short text contexts [2], called the self-resource-based approach. While the other is to extend the feature space by external sources, called the external resource-based approach, and it can be divided into four categories [3], [4]. The first one is to use the link information existing among short texts [5], called the link-based approach. The second one is to directly fetch external text (such as Web search snippets) to expand the short text [6], called the Web search-based approach. The third one is to fetch extra semantic information in knowledge bases such as WordNet [7] and Wikipedia [8], called the taxonomy-based approach. The last one is to discover explicit or implicit topics using external resources and then connect the short text through these topics, called the topic-based approach.

Among the above approaches, the self-resource-based approach can spare the time cost in the feature extension compared to that using external resources, but most of the studies in this category are application dependent or algorithm dependent. They still suffer from the severe data sparsity problem, such as sparse word co-occurrence patterns in individual document. It is hence hard to outperform those approaches based

Manuscript received December 19, 2016; revised April 18, 2017, June 17, 2017, and August 25, 2017; accepted August 29, 2017. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1000901, in part by the U.S. National Science Foundation under Grant IIS-1613950, in part by the Natural Science Foundation of China under Grant 61503112, Grant 61673152, and Grant 61503114, and in part by the Natural Science Foundation of Anhui Province under Grant 1708085QF142. A preliminary version of this paper was published in the Proceeding of International Conference on Data Mining (ICDM'16), pp. 1009–1014, 2016. This paper was recommended by Associate Editor S. Ozawa. *(Corresponding author: Yuhong Zhang.)*

P. Li, L. He, H. Wang, X. Hu, Y. Zhang, and L. Li are with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China (e-mail: peipeili@hfut.edu.cn; luhe@mail.hfut.edu.cn; haiyanwang@mail.hfut.edu.cn; jsjxhuxg@hfut.edu.cn; zhangyh@hfut.edu.cn; lilei@hfut.edu.cn).

X. Wu is with the School of Computing and Informatics, University of Louisiana at Lafayett, Lafayett, LA 70504-3694 USA (e-mail: xwu@louisiana.edu).

Digital Object Identifier 10.1109/TCYB.2017.2748598

on external resources. Regarding the external resource-based approaches, they can improve the classification accuracy with the help of more information to understand the short texts, but they still face the following weaknesses. First, a major problem with the link-based approach is that link information is not always available, thus the graph-based method built on the link is not applicable for all short text classification scenarios. The Web search-based approach requires interaction with a search engine which has high communication overhead and high index costs, and it is not suitable for online applications. The taxonomy- and topic-based approaches using explicit predefined topics/taxonomy relax the dependence on search engines, but they heavily depend on the completeness of the underlying taxonomy and the external corpora. More specifically, a popular taxonomy like WordNet [7] does not have the adequate coverage as it cannot keep up with the development of new terms and phrases everyday. Though it is easy to collect a huge text corpus (such as Wikipedia), its adaptability can be an issue since the predefined topics and taxonomy may not be available for certain applications. In addition, all of the aforementioned approaches are batch algorithms, thus, they are not suitable for short text stream classification due to lower efficiency.

In this paper, we propose a new feature extension approach for short text stream classification using a large scale, general purpose semantic network obtained from a Web corpus. Our main contributions of this paper are as follows.

A. First, Our Approach Produces Higher Classification Accuracy

Unlike the external resource-based approaches, we use an open semantic network called Probase[1] [9] instead of other popular taxonomy knowledgebases (such as Wikipedia) as the external resource. This is because Probase is one order of magnitude larger than Wikipedia in terms of the number of hypernym-hyponym relations. It can cover any two known noun-based multi-word expressions. We call them as terms in common. According to the taxonomy knowledge in Probase, we first introduce more semantic contexts based on the senses of terms to make up of the data sparsity, and then disambiguate all terms in short texts to reduce the impact from irrelevant senses.

B. Second, Our Approach Can Detect the Drifts of Topics Hidden in Short Text Streams

To track topic drifts hidden in short text streams, we propose a topic drifting detection method based on the sense distribution of terms. It is capable of capturing the drifts of topics in short text streams effectively and efficiently. Contrary to the classification-error-based concept drifting method, we use concept-based clusters to represent data distributions of each chunk, and then detect the hidden topic drifts in terms of the difference between concept-based clusters in adjoining two data chunks.

[1] http://research.microsoft.com/en-us/projects/probase/release.aspx

C. Finally, Our Approach Is Lightweight and Scalable

Compared to most of existing short text classification approaches, our approach is built on an incremental ensemble classification model. It is more efficient and scalable compared to several state-of-the-art algorithms for short text classification.

The rest of this paper is organized as follows. Section II summarizes related work. Section III presents the details of our approach. Section IV provides the experimental analysis. Finally, we conclude in Section V.

II. Related Work

Researchers mainly focus on addressing the challenges in short text classification from two directions, that is, one depends on the self-resources such as rules or statistical information hidden in the current short text contexts, called the self-resource-based approach, while the other depends on the external sources, called the external resource-based approach. More details are as follows.

A. Self-Resource-Based Approach

Several representative works of the self-resource-based approach are summarized below. Yuan *et al.* [10] tried to optimize the naïve Bayes (NB) algorithm to make it adaptable to the sparse data to improve the accuracy. Wang *et al.* [11] proposed a novel method to model short texts based on semantic clustering and convolutional neural network using pretrained word embeddings. Gao *et al.* [4] introduced a structured sparse representation classifier to effectively classify short texts. Haddoud *et al.* [12] proposed 80 metrics never used for the term-weighting problem for text classification. Bicalho *et al.* [13] proposed a topic model for short texts by creating larger pseudo-document representations from the original documents using word co-occurrence and word vector representations. Doulamis *et al.* [14] exploited pairwise similarities and intercorrelated words based on fuzzy time feature series to detect events in Twitter microblogging. As compared to those using external resources, this kind of methods present the superiority in efficiency, but most of them are application/algorithm dependent, and they still suffer from the data sparsity problem.

B. External Resource-Based Approach

We can divide the external resource-based approach into the following four categories [3], [4].

1) *Link-Based Approach:* It relies on additional link information to construct a graph of texts. For example, Wang *et al.* [5] proposed a graph-based method using posts (posted by the same author or two friends) for tweet classification. Thus the classification model contains a regularization term that restricts the difference between posts from connected authors to be small. Unfortunately, a major weakness in this kind of methods is that it is hard for the graph-based method to apply in all short text classification scenarios, because the link information cannot be always available.

2) *Web Search-Based Approach:* To enrich the short text, researchers use the search engines by treating short text as

a query and submitting it to a search engine. The search results, presented in terms of Web page titles and snippets, are widely used to enrich short texts. Main works are below. Bollegala *et al.* [6] proposed a semantic similarity computation method between words using page counts and snippets from Web search. Xu *et al.* [15] studied the continuous similarity search for evolving queries using pruning strategies and the MinHash technique. The above methods use the search engines to enrich the short text, which provides more information to understand the short text. But it is not applicable in the tackling of large-scale data sets, due to the high time cost and the heavy dependency on the quality of search engines.

3) *Taxonomy-Based Approach:* Most methods in this direction use explicit taxonomy in extra knowledgebase or corpora such as WordNet and Wikipedia. These corpora especially for Wikipedia have rich predefined taxonomy and human labelers assign thousands of Web pages to each node in the taxonomy. Such information can greatly enrich the short text. Main works are below. Zhai *et al.* [16] presented a semantic similarity-based short text classification method using WordNet and the Brown Corpus. Shirakawa *et al.* [8] proposed a Wikipedia-based semantic similarity measurement method for real-world noisy short texts. Yu *et al.* [17] proposed to enrich short texts with concepts and co-occurring terms extracted from Probase, and then introduced a simplified deep learning network consisting of a three-layer stacked auto-encoders for semantic hashing.

4) *Topic-Based Approach:* This method is to use the implicit topics (or concepts) mined from the external resources to expand the short texts. Main works are below. Phan *et al.* [18] mined implicit topics from the Wikipedia's texts with latent Dirichlet allocation (LDA) model and then used the topics as appended features to expand the short text. Wang *et al.* [19] proposed a short text categorization method using the topic model built from Wikipedia and an integrated classifier composed of maximum entropy and support vector machine (SVM) classifiers. Bouaziz *et al.* [20] proposed a random forest (RF) based approach that combines data enrichment with the introduction of semantics using Wikipedia and LDA. Cheng *et al.* [2] presented a refined LDA algorithm for biterm topic model (BTM). BTM learns topics by directly modeling the generation of word co-occurrence patterns in the corpus, making the inference effective with the rich corpus-level information from Wikipedia. Zuo *et al.* [21] proposed a pseudo-document-based topic probabilistic model for short texts. Xuan *et al.* [22] proposed an innovative graph topic model for classification on documents and chemical formula. Zuo *et al.* [23] presented a word co-occurrence network-based model using LDA and Wikipedia corpus to tackle the sparsity and imbalance of short texts simultaneously.

Aforementioned taxonomy/topic-based approaches relax the dependence on search engines, but they require a completely underlying taxonomy or external corpus. However, the popular taxonomy like WordNet does not have the adequate coverage. The other taxonomies such as Wikipedia can easily collect a huge text corpus, but it is probably not applicable for all applications because of predefined topics and taxonomy unavailable. In addition, all of the above works classify short

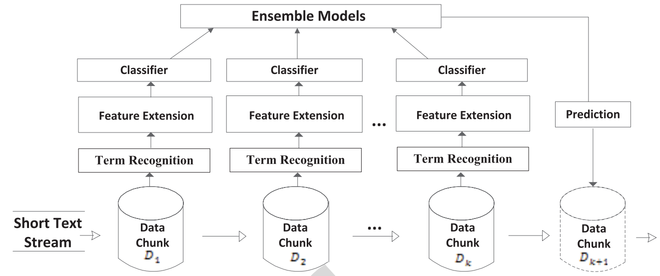


Fig. 1. Framework of our approach.

texts use batch processing. Thus, it is a challenge in the handling of short text stream classification due to the efficiency. Our proposed approach also belongs to the external taxonomy knowledge-based approach, but the difference lies that it is more scalable and effective contrary to the above works. This is because: 1) the knowledge we use was acquired from the entire Web; 2) our approach can reduce the noise impact from irrelevant senses of terms using the disambiguation; and 3) our approach can distinguish the topic changes hidden in the short text stream using the concept cluster-based drifting detector.

III. Our Short Text Stream Classification Approach With Topic Drifting Detection

In this section, we first introduce the formulation of our short text stream classification with topic drifting detection, and then give the technical details in our approach.

A. Problem Formulation

The problem of our short text stream classification is formalized below. Given a short text stream D, we can divide it into N data chucks, denoted as $D = \{D_1, D_2, \ldots, D_N\}(N \rightarrow\propto)$, where each data chunk consists of $|D_i|$ short documents, denoted as $D_i = \{d_1, d_2, \ldots, d_{|D_i|}\}$ $(1 \le i \le N)$, and each document can be vectorized as $d_j = \{(v_j, y_j)|v_j \in E, y_j \in Y\}(1 \le j \le |D_i|)$, where $E = R^M$ indicates the domain of the feature space, Y indicates the set of document classes with L labels, and M is the dimensionality. Short text stream classification aims to train a dynamic classifier $f : E_{\sum_{D_i}} \rightarrow Y$ that maps a feature vector to a set of labels, which can adaptively adjust varying with the seen short texts and the occurring topic drifts.

Fig. 1 shows the framework of our approach for this problem. It is built on an ensemble model consisted of K base models, denoted as $\lambda = \{\lambda^1, \lambda^2, \ldots, \lambda^K\}$, such that to each yet-to-come document d, the ensemble λ can assign a class label y^* which satisfies

$$y^* = \text{argMax}_{y\in Y} P(y|d, \lambda) \tag{1}$$

where $P(y|d, \lambda)$ is the weighted average of all K base models, denoted as $P(y|d, \lambda) = \sum_{i=1}^{K} \omega_i P(y|d, \lambda^i)$ and $\omega_i = 1$. In our approach, we represent each document d as a feature space with a set of terms and non-noun words, where terms indicate concepts and instances extracted in the short text using the Probase knowledgebase. According to the statistical results, the minimum rate of short texts containing terms in our experiments is also up to 95%, we hence only use the feature space of terms to represent each short text for simplicity, denoted as $V_d = \{T\} = \{t_i | 1 \le i \le n_t\}$, where n_t is the size of terms. To achieve the feature space of V_d, we require recognizing terms in short texts. It is relevant to two important techniques of term recognition and feature extension. Before giving the details of these two techniques, we first introduce the knowledgebase preliminary.

B. Knowledgebase Preliminary

To extend the features of short text streams, we need to use an open semantic network called Probase [9] to recognize the hidden terms. It has the following properties.

First, contrary to the tree structured taxonomy in popular knowledgebases (e.g., Wikipedia), Probase is a network in which an instance or concept probably have many super-concepts. It provides probabilistic isA knowledge for 2.7 million concepts, and it is one order of magnitude larger than Wikipedia in terms of the number of isA relations. All isA relationships are harvested from 1.68 billion Web pages and two years worth of Microsoft Bings search log using syntactic patterns, such as the Hearst's [24] patterns. For example, "...Asian countries such as China, ..." serves as an evidence that China is of type Asian country. In this case, "China" is an instance (namely hyponym) while "Asian country" is a concept (namely hypernym). We call the instances and concepts in common as terms here. For a concept/instance pair $\langle c, e\rangle$, it provides two *typicality* scores: $P(e|c)$ and $P(c|e)$. It is known as typicality because, for example, $P(cat|mammal) > P(whale|mammal)$ because *cat* is more typical than *whale* as a *mammal*. Typicality score is derived below, $P(e|c) = N(c,e)/N(c)$, where $N(\cdot)$ indicates the occurrences of given terms or term pairs in Hearst extractions.

Second, it provides the synsets, because a single term may have many surface forms such as "hp" and "hewlett packard." The set of all synsets provides a mapping between any Probase term, to its synset and hence all the other terms in that synset. Finally, many concepts in Probase are similar to each other, such as "music star" and "pop star." We use concept clusters to gather similar concepts together, by using a k-medoids clustering algorithm [25]. One concept cluster can represent a sense or a general topic, recognized with its center concept. For example, for the cluster centered around company, most of its members are highly related to company, such as software company and technology company.

C. Knowledgebase-Based Term Recognition

To recognize terms, the Stanford natural language processing tool is first used to obtain bag-of-words given a document after preprocessing, such as parsing and removing stop words. And then the backward maximum matching (BMM) method [26] is applied to efficiently find all terms using Probase. For example, two short texts are given below.

1) This is the **compilation** of the new york times **adult best seller** **lists** that are on the hawes publications **Website**.
2) Apple had *agreed* to **license** certain parts of its **GUI** to Microsoft for **use** in Windows 1.0."

According to the *BMM* method, we can only get a term with the longest coverage namely "new york times." On the other hand, we can get all terms marked in underline and non-noun words marked in italic font (such as "agree") according to the Probase knowledgebase. Meanwhile, we further distinguish the type of terms according to the rule defined in (2), where r is a ratio threshold, $I(t)/C(t)$ indicates the set of instances/concepts that belong to the term t as a/an concept/instance, respectively, $|I(t)|/|C(t)|$ indicates the size of the corresponding set, respectively, and freq(c)/freq(e) indicates the frequency of the concept c and the instance e in Probase, respectively. In this case, we can divide terms into concepts (marked in underline and in bold font) and instances (marked in underline only)

$$\text{type}(t) = \begin{cases} \text{concept} & \text{if } r \ge 1 \\ \text{instance} & \text{otherwise} \end{cases} \tag{2}$$

s.t., $r = \sum_{e \in I(t)} (\text{freq}(e) \cdot |I(t)|) / (\sum_{c \in C(t)} (\text{freq}(c) \cdot |C(t)|) + 1)$.

Finally, we can get the feature space of a short text as $V_d = \{T\} = \{C, I\}$, where C indicates the set of concepts with the size of n_c concepts, namely $C = \{c_1, c_2, \ldots, c_{n_c}\}$, and I indicates the set of instances with the size of n_e instances, namely $I = \{e_1, e_2, \ldots, e_{n_e}\}$. Therefore, the corresponding feature vector can be represented as $\mathcal{I}_d = \{\mathcal{I}_C, \mathcal{I}_E\}$, s.t., $\mathcal{I}_C = \langle w_1, w_2, \ldots, w_{|n_c|}\rangle$ and $\mathcal{I}_I = \langle w'_1, w'_2, \ldots, w'_{|n_e|}\rangle$, where $w_i (1 \le i \le n_c)$ and $w'_i (1 \le i \le n_e)$ indicates the Tf-idf value of the concept c_i and the instance e_i in the given document, respectively.

D. Feature Extension Based on Concepts

To make up the data sparsity, we use the semantic concepts to extend the feature space of the short text here. Fig. 2 illustrates how to extend the feature space of each original short text. From Fig. 2, we can see that according to the term recognition mentioned above, we can first get the term set given a short text of "Apple had agreed to license certain parts of its GUI to Microsoft for use in Windows 1.0," and then we label the type of each term using type checking defined in (2). In terms of all labeled concepts and instances

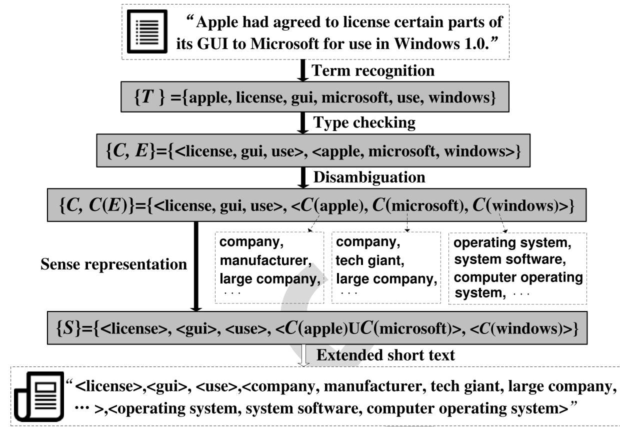


Fig. 2. Illustration to feature extension.

(denoted as $\{C,\ I\}$), we will induce the dominant sense each instance belongs to (denoted as $C(I)$), namely disambiguation, for example, the dominant sense of instance "apple" is related to the concept "company" in this short text due to the occurrence with "microsoft." Furthermore, we will represent the feature space using concept clusters of terms as the senses hidden in the short text, denoted as S, and then we can get the final feature space extended from the original short text as shown in Fig. 2. More details of techniques such as the disambiguation and the sense representation using concept clusters are below.

First, we should determine which dominant concepts the above recognized instances in I belong to. This is because in the Probase knowledgebase, we know that the instances probably have ambiguous concepts (namely senses). For example, *apple* belongs to at least two concepts like *company* and *fruit*. According to the above term recognition, we can preliminarily distinguish the types of terms, namely the concepts and the instances, but we cannot disambiguate the senses of instances, such as the concept *apple* belongs to as shown in Fig. 2. Thus, it is necessary to determine which concept the instance belongs to in terms of the context in the current short text. Details are below.

We introduce the entropy-based method [26] as shown in (3) to judge which recognized instances are ambiguous,

$$H(e) = -\sum_{cl_x \in CL_e} P(cl_x|e) \cdot \log_2 P(cl_x|e) \tag{3}$$

where $H(e)$ indicates the entropy value given an instance, $P(cl_x|e) = \sum_i^n w_{c_i} P(c_i|e)$, $cl_x = \{c_i|1 \le i \le n\}$ indicates a cluster of concepts that the given instance e belongs to, $w_{c_i} = \sum_{e_j \in I} P(c_i|e_j) \times idf(e_j) \times idf(c_i)$ aggregates the weights of votes to it from all its instances in the instance set I from Probase, and $CL_e = \{cl_x|1 \le x \le m\}$ indicates the concept cluster set with m concept clusters given the instance e. We use a k-medoids clustering algorithm to generate concept clusters, in this case, a concept cluster can represent one sense, recognized with its center concept. For example, for the cluster centered around company, most of its members are highly related to company, such as firm, software company, large company and manufacturer. More details refer to our previous work in [25]. That is, an instance with a high entropy value $H(e)$ tends to have high uncertainty of cluster distribution.

To determine the dominant sense of an ambiguous instance e in the given short text, we compute the similarity between each concept cluster of e and the concept cluster of all unambiguous terms (including the instances and the concepts) using the sense detection method proposed in [26]. Fig. 3 shows an example of how to detect the sense of ambiguous instance *apple* by the unambiguous instance *microsoft*, in which "concept clustering." From this figure, we can see that *apple* has multiple senses, such as *fruit* and *company*, but we can disambiguate the sense of *apple* as *company* when it co-occurs with the unambiguous instance *microsoft*.

According to the above analysis, the feature space of each instance can be represented as the feature space of many concepts in the same sense. For example, the feature

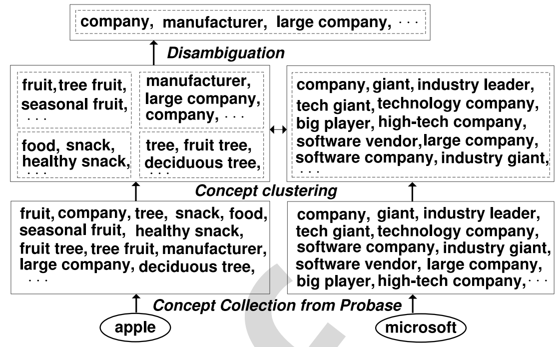


Fig. 3. Illustration to disambiguation.

space of *apple* is represented as the concept set of *company*, *manufacturer*, and *large company* in the company sense. Correspondingly, we can represent the feature space of the short text d as $V_d = \{C, C(I)\}$, where $C(I)$ indicates the concept clusters of instances in I, denoted as $C(I) = \{C(e_x)|1 \le x \le |n_e|\}$, and $C(e_x)$ indicates the concept set of the instance e_x in the same concept cluster (namely in a sense), denoted as $C(e_x) = \{c_i|1 \le i \le |e_x|\}$, and c_i is a concept[2] the instance e_x belongs to. Therefore, the corresponding feature vector of $C(I)$ can be represented as

$$\mathcal{I}_{C(I)} = \left\{\mathcal{I}_{C(e_x)}|1 \le x \le n_e\right\} \tag{4}$$

s.t., $\mathcal{I}_{C(e_x)} = \langle w_1, w_2, \ldots, w_{|e_x|}\rangle$, where $w_i = p(c_i|e_x) \cdot w_{e_x}$, $p(c_i|e_x)$ is the typicality score for e_x and concept c_i ($c_i \in C(e_x)$, $1 \le i \le |e_x|$), that is, how typical c_i is among all the concepts e_x belongs to, and w_{e_x} indicates the Tf-idf value of the instance e_x in the given short text.

After disambiguation, we further reorganize the concept sets according to the senses of terms to represent the feature space of each short text. In this paper, we utilize the concept clusters to represent the hidden senses. More specifically, we first get the clusters of concepts in C using the concept clusters mentioned in [25]. Second, we merge the concept clusters in the same sense. For example, the dominant concept clusters of *apple* and *microsoft* in Fig. 2 share the same sense *company*, we can merge both concept clusters into one, denoted as $C_1 \bigcup C_2$ given two concept clusters C_1 and C_2. In this case, we can represent the feature space of each short text using the senses as $V_d = \{S\} = \{S_i|1 \le i \le k\}$, namely the concept set in $\{C,\ C(I)\}$ is reorganized into k concept clusters, s.t., $S_i = \{c_j^i|1 \le j \le |S_i|\}$, where each concept cluster S_i indicates a sense, containing some concepts in $\{C,\ C(I)\}$, and c_j^i is the jth concept in the ith concept cluster S_i with the size of $|S_i|$. Correspondingly, the feature vector of this short text can be finally represented as

$$\mathcal{I}_d = \mathcal{I}_S = \{\mathcal{I}_{S_i}|1 \le i \le k\} \tag{5}$$

s.t., $\mathcal{I}_{S_i} = \langle w_{i1}, w_{i2}, \ldots, w_{i|s_i|}\rangle$, where $w_{ij}(1 \le j \le |s_i|)$ indicates the weight of the jth concept in S_i, it is defined as same as that of w_i mentioned in (4).

[2]Noisy concepts with only one occurrence are filtered.

Algorithm 1 Concept-Based Short Text Clustering

Input: D_i: the i^{th} data chunk; T: the maximum iteration count (eg., 10^4); σ: the divergence threshold between clusters (eg., 10^{-3}); τ: the divergence threshold between cluster centers (eg., 10^{-10});

Output: L clusters $\{K_1, K_2, \ldots, K_L\}$;

1: Initialize the iteration time: $t = 0$;
2: Initialize the cost variables: $cost = oldCost = 0$;
3: Generate an initial center set $G^t = [g_1^t, g_2^t, \ldots, g_L^t]$;
4: Assign each short text d_j to a cluster K^* with a center g^* satisfying (7) and update the cost, namely $cost += dist(g^*, d_j)$;
5: Update cluster centers in G^{t+1} using (8);
6: $\Delta_{mean} = \sum_{i,j=1}^{L} dist(g_i^{t+1}, g_j^t)$;
7: $\Delta_{cost} = |cost - oldCost|/(oldCost + \epsilon)$ $(\epsilon = 10^{-10})$;
8: $oldCost = cost$;
9: **if** $\Delta_{cost} > \sigma$ and $\Delta_{mean} > \tau$ and $t < T$
10: Let $t = t$+1 and go to Step 3;
11: return clusters $\{K_1, K_2, \ldots, K_L\}$;

E. Concept Cluster-Based Topic Drifting Detection

Our concept cluster-based topic drifting detection method is proposed in detail. According to the above analysis, the feature space of senses is first obtained for each data chunk, where senses indicate semantic concepts. Second, the refined k-means clustering algorithm is adopted to find clusters of short texts in the current data chuck. Because k-means is more suitable for numerical attributes, and it is simple and fast in the handling of the small scale of data chunks. In this case, the label distribution of this data chuck can be represented by concept clusters. Finally, we compute the distances between cluster centers in the adjoining two data chunks using the cosine function [denoted as $\cos(\cdot)$] to judge the topic drifts. Technical details are below.

1) *Clustering Algorithm:* The semantic distance between two short texts d_i and d_j is first defined as

$$\text{dist}(d_i, d_j) = 1 - \cos(\mathcal{I}_{d_i}, \mathcal{I}_{d_j}) = 1 - \cos\left(\mathcal{I}_{S_{d_i}}, \mathcal{I}_{S_{d_j}}\right), \text{ s.t.}$$

$$\cos(\mathcal{I}_{S_{d_i}}, \mathcal{I}_{S_{d_j}}) = \frac{\sum_{S_x \in S_{d_i} \bigcap S_{d_j}} \left(\mathcal{I}_{S_x}^{d_i} \cdot \mathcal{I}_{S_x}^{d_j}\right)}{\sqrt{\Sigma_{S_x \in S_{d_i}} \left\|\mathcal{I}_{S_x}\right\|_2^2} \cdot \sqrt{\Sigma_{S_y \in S_{d_j}} ||\mathcal{I}_{S_y}||_2^2}} \tag{6}$$

where $\mathcal{I}_{S_x}^{d_i} \cdot \mathcal{I}_{S_x}^{d_j} = \sum_k w_{xk} \cdot w'_{xk}$ $(w_{xk} \in \mathcal{I}_{S_x}^{d_i}$ and $w'_{xk} \in \mathcal{I}_{S_x}^{d_j})$.

A modified k-means clustering algorithm is used to partition short texts by their feature distributions. First, initial centers are randomly selected by the label distributions of short texts given a data chuck D_k, which is set to $G^0 = \{g_1^0, \ldots, g_L^0\}$ at iteration 0, where L is the label count. With L centers in the tth iteration, each candidate short text $d_i \in D_k$ is assigned to its closest center $g^* \in G^t = \{g_1^t, \ldots, g_L^t\}$, namely a center g^* with the minimum semantic distance from d_i as shown in the following equation:

$$g^* = \underset{g_j^t \in G^t}{\text{argmin}}\, \text{dist}\left(d_i, g_j^t\right). \tag{7}$$

When all candidate short texts are assigned to the corresponding clusters, the center is updated with the most centrally located point (virtual short text) in each cluster. To find such a center, the average distance of a cluster K_i is evaluated in (8),

$$g_i^{t+1} = \sum_{d_x \in K_i} \frac{\mathcal{I}_{d_x}}{|K_i|} = \sum_{d_x \in K_i} \frac{\mathcal{I}_{S_{d_x}}}{|K_i|} = \sum_{d_x \in K_i} \sum_{S_y \in S_{K_i}} \frac{\mathcal{I}_{S_y}}{|K_i|} \tag{8}$$

where S_{K_i} indicates the sense space in the current cluster K_i, namely $S_{K_i} = \{S_{d_x} | d_x \in K_i\}$. The clustering process iterates until the convergence condition is met. Algorithm 1 shows the framework of our modified k-means clustering algorithm on short texts.

2) *Distance Computation Between Concept Clusters:* According to the above analysis, each data chunk consists of several concept clusters, where a cluster indicates a sense represented by several concepts hidden in short texts. In other words, a data chunk can be represented as the sense distributions of concepts (denoted as $\mathcal{I}_{S_{D_x}}$). Correspondingly, we use the divergence between the sense distributions in the adjoining two data chunks to detect the hidden topic drifts, where the divergence is defined in

$$\begin{aligned} D_{\cos}(\mathcal{I}_{D_x}, \mathcal{I}_{D_y}) &= 1 - \cos\left(\mathcal{I}_{D_x}, \mathcal{I}_{D_y}\right) \\ &= 1 - 1/|S_{D_x}| \cdot \sum_{i=1}^{|S_{D_x}|} \text{Max}_{j=1}^{|S_{D_y}|} \cos\left(\mathcal{I}_{S_{K_i}}, \mathcal{I}_{S_{K_j}}\right) \end{aligned} \tag{9}$$

s.t., $\mathcal{I}_{D_x} = \{\mathcal{I}_{S_{K_i}} | 1 \le i \le |S_{D_x}|\}$, $\mathcal{I}_{D_y} = \{\mathcal{I}_{S_{K_j}} | 1 \le j \le |S_{D_y}|\}$, where $\cos(\mathcal{I}_{S_{K_i}}, \mathcal{I}_{S_{K_j}})$ is computed as similarly as shown in (6). $|S_{D_x}|$ and $|S_{D_y}|$ indicates the size of senses in data chunks of D_x and D_y, respectively. Considering the noisy impact in the topic drifting detection, we divide the cases of topic drifts into the following three categories according to two thresholds of α and β, namely: 1) if $D_{\cos}(\mathcal{I}_{D_x}, \mathcal{I}_{D_y}) \in [0, \alpha]$, no topic drift; 2) if $D_{\cos}(\mathcal{I}_{D_x}, \mathcal{I}_{D_y}) \in (\alpha, \beta)$, noisy impact; and 3) if $D_{\cos}(\mathcal{I}_{D_x}, \mathcal{I}_{D_y}) \in [\beta, 1]$, topic drift.

F. Ensemble Classifier Modeling and Prediction

To predict a yet-to-come short text, we require building classifiers in terms of the extended feature space mentioned above. In this paper, we build the model of concept clusters on each data chunk as a classifier denoted as $\lambda^i = \{K_j^i | 1 \le j \le L\}$, correspondingly we can get an ensemble classifier based on concept clusters built on K data chunks, denoted as $\lambda = \{\lambda^1, \lambda^2, \ldots, \lambda^K\}$. In terms of the ensemble model, we can predict each short text below. Given a testing short text, we first extend semantic concepts of terms hidden in this short text and use the senses (concept clusters) of terms to represent the feature space. Second, we find the K nearest concept clusters by comparing the semantic distance between the current short text and the centers of concept clusters in recent seen K data chunks. Third, we can predict the current short text by the label with the maximum probability, denoted as $L(d) = \text{argMax}_{y_j \in Y} \Sigma_{i=1}^{K} P(y_j | K_j^i)$ $(1 \le j \le L)$.

G. Analysis of Time and Space Complexities

The time consumption of prediction in our approach is similar to all baselines because it is mainly linearly relevant to the number of testing instances, while the training time consumptions are significantly different. More precisely, the training cost in our approach mainly consists of term recognition, feature extension and concept cluster generation. Because the size of a sliding window used in our approach containing K data chunks is a constant, the time complexity of our approach can be represented as $O(|D_i| \cdot |W| + |I| \cdot |C(I)| \cdot |D_i| + L \cdot |D_i|^2)$, where $|D_i|$ indicates the size of a data chunk, $|W|$ indicates the average number of words in a text, $|I|$ indicates the number of instances contained in each text, $|C(I)|$ indicates the number of concept clusters all instances belong to, and L indicates the number of cluster centers. As compared with all baselines, first, time complexities of S_SVM-, NB-, and Spegasos-based approaches are in direct proportion to the size of all texts and the length of the text, denoted as $O(|D| \cdot |W|)$, due to $|D| \gg |D_i|$, thus, they will present the disadvantage in the time cost as the scale of training data increases. Second, the time cost of Wikipedia resource-based algorithms mainly depends on the acquisition of external resources, and the time complexity is usually proportional to the size of training data or even more. Thus, it is inferior to our approach. Third, time complexities of OzaBagASHT- and KNN+PAW+ADWIN-based approaches can be represented as $O(C \cdot T \cdot n' \cdot |W|)$ and $O(k' \cdot n'^2)$, respectively, where T is the iteration count, C indicates the count of base classifiers, n' indicates the size of a sliding window, and k' indicates the number of nearest neighbors. Due to usually $n' >> |D_i|$, the former two approaches are also inferior to ours.

On the other hand, the space complexity in our approach is proportional to the size of a data chunk and all cluster centers containing different concepts, denoted as $O(|D_i| \cdot A + L \cdot B)$, where A and B indicates the average space cost of an extended instance and a cluster center, respectively. Because L is a constant, the main space cost depends on the size of a data chunk. As compared with all baseline algorithms, first, space complexities of S_SVM-, NB-, and Spegasos-based approaches and Wikipedia resource-based algorithms are mainly relevant to the size of a short text stream, denoted as $O(|D| \cdot A')$ and I$(|D| \cdot A)$, respectively, where A' indicates the average space consumption of each instance without feature extension. In general, $A' < A$ is satisfied, but due to $|D| \gg |D_i|$, our approach will present the prominent advantage in the space consumption. Second, the space complexities of OzaBagASHT- and KNN+PAW+ADWIN-based algorithms can be represented as $O(C \cdot |D| \cdot A')$ and $O(k' \cdot n' \cdot A')$, respectively. It is also obvious to get that our approach presents a lighter space consumption due to $|D| \gg |D_i|$ and $n' > |D_i|$.

H. Scalability Analysis

We provide a brief study about how our approach could perform with a much bigger short text stream in terms of volume and topic diversity collection here. Regarding the high-volume of a short text stream, our approach is influenced from the sliding window mechanism. The arrived short text stream is divided into small data chunks and a sliding window used in our approach contains K data chunks. Each classifier is incrementally built on a data chunk to generate an ensemble model. To adapt to a larger scale of a short text stream, the ensemble model will update the worst classifier with a new one by the time stamp or the topic change. Regarding the topic diversity in a short text stream, we apply the sense distribution-based topic drifting detection method to distinguish topic changes from noisy data, where senses are represented by the concept clusters. To reduce the impact from the sparsity of short texts and noisy data, we use concepts obtained from the Probase knowledgebase to extend the feature space. According to the above analysis, time and space consumptions of our approach mainly depend on the size of a sliding window. Our approach is therefore scalable, and it can be implemented in a parallel framework using Hadoop or Spark, because of the ensembling framework in a sliding window.

IV. Experiments

In this section, we first outline the experimental setup, and then compare the effectiveness of our approach with several state-of-the-art approaches in the topic drifting detection and in the classification accuracy. Finally, we evaluate the efficiency of our approach.

A. Experimental Setup

Benchmark Data Sets We use three well-known benchmark short text data sets as follows.

1) *Snippets* [18]: Web search snippets consist of three parts: 1) a URL; 2) a short title; and 3) a short text description. They were selected from the results of Web search transaction using predefined phrases of different domains. For each query phrase put into Google search engine, the top 20 or 30 ranked Web search snippets were collected. Then the class label of the collected search snippets was assigned as the same as that of the issued phrase.
2) *News:* News is from TagMyNews Data Sets[3], which is a collection of data sets of short text fragments used for the evaluation of the topic-based classifier. It contains 32K English news extracted from RSS feeds of popular newspaper Websites (nyt.com, usatoday.com, and reuters.com) with seven categories. In our experiments, we extract the title-only, description-only and both of titles and descriptions, respectively, as three groups of data sets.
3) *Tweets:* Tweets provides about 400 K tweets with five categories [27]. The topic related to obama is obtained by conducting keyword filtering on a large Twitter data set used by [28]. The other four topics are acquired during November and December in 2012 via Twitter's keyword tracking API[4].

Table I summarizes the details of label distributions in the above benchmark data sets. In our experiments, we suppose

[3]http://acube.di.unipi.it/tmn-dataset/

[4]https://dev.twitter.com/docs/api/1.1/post/statuses/filter

TABLE I
DATA SETS USED IN THE EXPERIMENTS

Data set	Domain	#documents	#total
Snippets	Business	1500	12340
	Computer	1500	
	Culture-Arts-Ent	2210	
	Education-Science	2660	
	Engineering	370	
	Health	1180	
	Politics-Society	1500	
	Sports	1420	
News	Sport	8190	32604
	Business	5367	
	U.S.	4783	
	Health	1851	
	Sci&Tech	2872	
	World	6255	
	Entertainment	3286	
Tweet stream	arsenal	82000	408000
	blackfriday	70000	
	chelsea	86000	
	smartphone	74000	
	obama	96000	

documents with the same label indicate a topic. To simulate topic changing, we randomly generate a group of data sets with a fixed period of topic changing (e.g., CP = 500), namely it is changed from a topic to another one every 500 short texts. Meanwhile, the data set is added by $r\%$ noisy data (e.g., $r = 5$). Correspondingly, we can get five synthetic data sets. In our experiments, all data sets are partitioned into N data chunks to simulate the data streams, and each data chunk (D_i) contains 50 short texts. In the topic drifting detection, we detect the topic drift in every data chunk, called the checking period. In the other hand, to investigate the performance of our topic detection method in a real data environment, we use Twitter's keyword tracking API to collect about 300 K short texts during November 26 to December 25 in 2012 with four classes (arsenal, blackfriday, chelsea, and smartphone) as a real data set, called Tweets_R. In this data set, all short texts are sorted by time stamps, which are partitioned by an hour in a day. To simulate the real environment with hybrid topics, gradual or abrupt drifts and noise data, the sequence of all short texts with the same time stamp is generated randomly, that is, there are many topics occurring during an hour, and each topic contains different sizes of texts. Meanwhile, short texts with irrelevant topics are added as noisy data and the noise rate is set to 10%. Fig. 4 illustrates the data distribution of topics over the first 50 data chunks.

1) *Baseline Approaches:* In our experiments, we will investigate the effectiveness of our approach[5] in the following dimensions. One is the performance in the topic drifting detection. We select nine state-of-the-art concept drifting detectors in data streams as the competing algorithms, and details are shown in Table II. To evaluate the performance of the drifting detectors, we introduce four data stream classifiers as base classifiers in the topic drifting detection, including NB, Spegasos, KNN+PAW+ADWIN, and OzaBagASHT as

[5]https://github.com/peipeilihfut/ShortTextStreamClassification

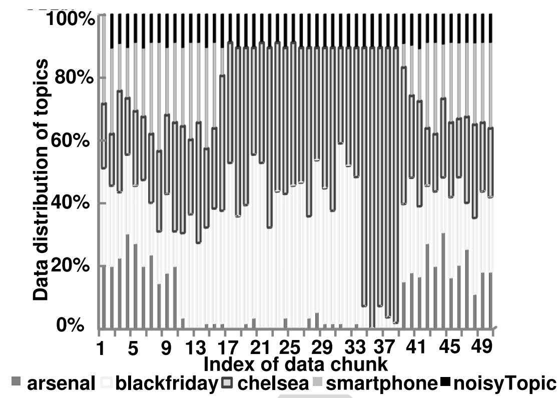


Fig. 4. Illustration to data distribution of topics in Tweets_R.

shown in Table III. All competing drifting detectors and base classifiers are from the open source experimental platform of massive online analysis (MOA) [29]. It is a software environment for implementing algorithms and running experiments for online learning from data streams. The other dimension is the classification performance. Thus, we select eight classification approaches as the baselines, including the mentioned four base classifiers for data streams and four well-known short text classification approaches. All competing classification methods involved in this section are summarized in Table III. All data stream classification approaches are from the MOA open source, while S_SVM is from the open library for short text classification[6] and the topic model LDA is implemented based on Gibbs sampling using the open source in Java[7]. To handle short text streams, a sliding window is used to train models in all short text classification approaches.

2) *Evaluation Measures:* In the topic drifting detection, we introduce the prequential evaluation [40] using fading factor (e.g., 0.995) to monitor the topic changing in the short text stream classification. Because it is more suitable for concept drifting detection in data stream. Meanwhile, we use three evaluation measures for statistics of drifting detection.

1) *False Alarm:* The rate that false alarms occur in the drifting detection.
2) *Missing:* The rate of topics missed in the drifting detection.
3) *Delay:* The mean count of instances required to detect the drift after the occurrence of a topic drift.

In the classification, we use the incremental accuracy to monitor the classification performance in the short text stream. That is, the classifier is trained and updated on the ith data chunk and used to predict on the next data chunk. Thus, with the arriving of data chunks, we can get a set of accuracy predicted on each data chunk. In addition, thresholds used in the topic drifting detection are set to $\alpha = 0.2$ and $\beta = 0.5$. The number of data chunks in a sliding window is set to $K = 10$. All experiments are performed on an Intel Core 2 Duo 2.66-GHz PC with 4-GB physical memory, running Windows 7 Enterprise. All timing results are averaged over five runs.

[6]http://www.csie.ntu.edu.tw/cjlin/libshorttext/
[7]http://jgibblad.sourceforge.net/

TABLE II
TOPIC DRIFTING DETECTORS (EWMA: EXPONENTIALLY WEIGHTED MOVING AVERAGE)

Approach	**Description**
DDM [30]	Drift Detection Method
EDDM [31]	Early Drift Detection Method
ADWINChangeDetector [32]	Drift detection method based on adaptive sliding window
CusumDM [33]	Drift detection method based on Cusum (Cumulative Sum of Recursive Residual)
EWMAChartDM	Drift detection method based on EWMA control charts
OnePassSamplerR [34]	A refined sequential change detection model with reservoir sampling
PageHinkleyDM [35]	Drift detection method based on Page Hinkley Test
HDDM_A_Test and HDDM_W_Test [36]	Online drift detection method based on Hoeffding's bounds using the average as estimator and using the EWMA statistic as estimator respectively

TABLE III
COMPETING METHODS (LDA)

Category	**Approach**	**Description**
Data stream classifiers	Naïve Bayes	A single Naïve Bayes model based data stream classification approach
	Spegasos	Implements the stochastic variant of the Pegasos method [37]
	kNN+PAW+ADWIN	*k* Nearest Neighbor adaptive with Probabilistic Approximate Window and ADaptive sliding WINdow
	OzaBagASHT [38]	Online Bagging based on Adaptive Size Hoeffding Trees
Short text classifier	S_SVM [39]	Self-information based approach using SVM by Crammer and Singer
	Wiki+LDA+SVM [18]	Topic based approach using Wikipedia and the SVM classifier
	Wiki+LDA+MaxEnt [19]	Topic based approach using Wikipedia and the maximum entropy model
	Wiki+LDA+RF [20]	Topic based approach using Wikipedia and random forest classifier

B. Effectiveness

To evaluate the effectiveness of our approach, we first compare our topic drifting detection approach with nine state-of-the-art drifting detectors varying with four base classifiers. Second, we compare our approach with four data stream classification approaches and four short text classification approaches in the classification accuracy.

To conduct the performance analysis among all comparing approaches systematically, we employ Friedman test [41] widely accepted as the favorable statistical test for comparisons of multiple approaches over a number of data sets [35], [42]. Given k comparing approaches and N data sets, let r_i^j denote the rank of the jth approach on the ith data set. Let $R_j = (1/N)\sum_{i=1}^{N} r_i^j$ denote the average rank for the jth approach, under the null hypothesis, the following Friedman statistic F_F will be distributed according to the $F-$distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom, namely $F_F = [((N-1)\chi_F^2)/(N(k-1)-\chi_F^2)]$, s.t., $\chi_F^2 = [12N/(k(k+1))][\sum_{j=1}^{k} R_j^2 - ([k(k+1)^2]/4)]$.

We can get the values of F_F using our approach and the competing ones in the topic drifting detection and in the classification as shown in Table IV. We can see that at significance level $\alpha = 0.05$, the null hypothesis of "equal" performance among the competing algorithms is clearly rejected in terms of each evaluation metric, because the value of F_F is larger than the corresponding critical value. Consequently, we require proceeding with certain *post-hoc test* to further analyze the relative performance among the competing algorithms. As we are interested in whether the proposed approach achieves competitive performance against other competing approaches, we employ the *Bonferroni–Dunn test* to serve the above purpose by treating our approach as the control approach. Here, the difference between the average ranks of our approach and competing ones is compared with the following critical difference (CD): $\text{CD} = q_a\sqrt{k(k+1)/(6N)}$. For this test, we have $\text{CD} = 4.85$ ($k = 10$, $N = 6$, and $q_a = 2.773$) regarding the performance in the topic drifting detection, and $\text{CD} = 4.31$ ($k = 9$, $N = 6$, and $q_a = 2.724$) regarding the classification performance at significance level $\alpha = 0.05$, respectively. Accordingly, the performance between our approach and a competing one is deemed to be significantly different if their average ranks over all data sets differ by at least one CD. More details of experimental results are given as follows.

TABLE IV
SUMMARY OF THE FRIEDMAN STATISTICS F_F AND THE CRITICAL VALUE

<table>
<tr><th colspan="3">On the performance of topic drifting detection</th></tr>
<tr><th>Measure</th><th>F_F</th><th>Critical Value(α = 0.05)</th></tr>
<tr><td>False Alarm</td><td>2.92</td><td rowspan="3">$F_{\frac{\alpha}{2}}(k-1,(k-1)(N-1))$ = 2.45
$k=10,\ N=6$</td></tr>
<tr><td>Missing</td><td>5.46</td></tr>
<tr><td>Delay</td><td>3.41</td></tr>
<tr><th colspan="3">On the performance of classification</th></tr>
<tr><th>Measure</th><th>F_F</th><th>Critical Value(α = 0.05)</th></tr>
<tr><td>Accuracy</td><td>13.40</td><td rowspan="2">$F_{\frac{\alpha}{2}}(k-1,(k-1)(N-1))$ = 2.53
$k=9,\ N=6$</td></tr>
<tr><td>Time overhead</td><td>11.80</td></tr>
</table>

1) *Topic Drifting Detection:* In this section, we aim to evaluate whether the topic drifting detection technique in our approach could handle scenarios with topic drifts. In one dimension, we give the statistical test (namely CD diagrams) on the performance of topic drifting detection as shown in Fig. 5, where the average rank of each competing approach is marked among the axis, namely lower ranks to the right. In each subfigure, any competing approach whose average rank is within one CD to that of the best approach is interconnected

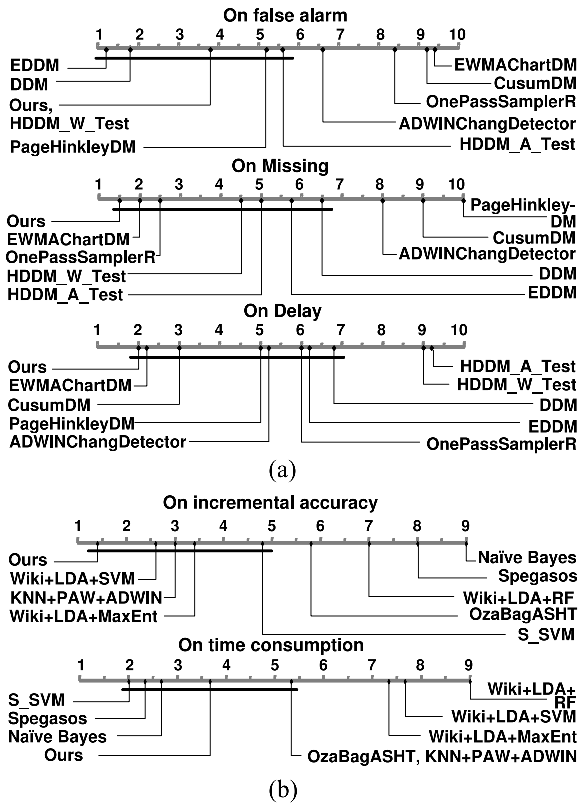


Fig. 5. Our approach against competing approaches on topic drifting detection and on classification with the *Bonferroni–Dunn test*. (a) Topic drift detection on data sets. (b) Classification on data sets.

with a thick line. Otherwise, any approach not connected with the best approach is considered to have significantly different performance between each other. Meanwhile, we summarize all topic drifting statistics with ranking in Table V. For clear clarification, we only give the average statistics of all competing drifting detectors over four base classifiers. From these experimental results, we can observe the followings.

First, our approach is the control one, which can beat all competing approaches on evaluation measures of delay and missing. More specifically, our approach is significantly superior to HDDM_A/W_Test on delay. This is because both drifting detectors of the latter use the non-weighted or weighted statistic as the estimator in the Hoeffding's bounds computation, it requires more statistical information of instances, which causes the long time delay. On missing, our approach can beat HDDM_A/W_Test, EWMAChartDM, and OnePassSamplerR, and all detectors present the significant advantage compared to CusumDM and ADWINChangeDetector. And on FAlarm, our approach is comparable to both detectors of HDDM_A/W_Test, which significantly outperform CusumDM, EWMAChartDM, OnePassSamplerR, and ADWINChangDetector. Reasons are analyzed below.

CusumDM uses sliding window schemes to compute test statistics in terms of a log likelihood ratio, this strategy is beneficial to report the topic drifts timely, but it will causes the higher Missing and FAlarm due to no enough informative statistics in the sparse and high-dimensional data with noise. ADWINChangDetector uses a variation of exponential histograms to limit the number of hypothesis tests done on a given window, but it suffers from the use of Hoeffding's bounds which greatly over estimates the probability of large deviations for distributions of small variance, namely it is sensitive to slow gradual changes. However, in the short text streams, most of topic changes are abrupt. Thus, these detectors relying on data compression/aggregation strategy present the higher missing and FAlarm. EWMAChartDM adopts the exponentially weighted moving average chart using Monte Carlo simulation (MCS) to detect concept changes, however, each arrived instance will trigger the MCS. While OnePassSamplerR adopts a more sensitive detection threshold with a reservoir sampling to manage data in the detection window. It is hence possibly to detect more topic drifts, which causes higher false alarms. However, HDDM_A/W_Test maintain both advantages of Hoeffding's bounds and the EWMA estimator, they can effectively detect abrupt and gradual drifts. But this conclusion is more suitable for low-dimensional and nonsparse data. When meeting short texts with high-dimensional and sparse data, the performance gets worse than ours. Because our approach introduces the external concept knowledge to make up of the data sparsity and considers the hidden semantics of short texts by building concept clusters. It is conducive to effectively detect topic drifts.

It is necessary to mention that DDM, EDDM, and PageHinkleyDM perform very well on false alarm as shown in Fig. 5(a), but they are built on the premise that at least a half of topic drifts are missing in the detection as shown in Table V. This is because DDM and EDDM identify a single cut point in the sequence of incoming values, by counting the number of errors or the error rate, while PageHinkleyDM considers the cumulated difference between the observed values and their mean, which is heavily impacted from the data distributions of attributes. Due to the sparsity and the high dimension of short texts, these methods cannot effectively detect topic drifts. Therefore, we can draw a conclusion that our approach outperforms all competing drifting detectors in the topic drifting detection, considering the tradeoff performance over all aforementioned evaluation measures.

In the other dimension, we give the details of experimental results in the topic drifting detection. Fig. 6 reports the topic detection curves based on the prequential error evaluation with factor fading in our approach and the best baseline. In these figures, we only draw parts of drifting detection curves on *News* and *Tweets(_R)* for clear investigation. And all drifting points are marked by the dotted lines on five synthetic data sets, because the concept changing period is fixed, namely CP = 500. Meanwhile, we select HDDM_W_Test as the baseline detector, because it performs best compared to other competing detectors. In the observation of tracking curves, we can see that if topics are changing, the drifting curves in HDDM_W_Test fluctuate more sharply than ours. And our approach can recover from each topic drift earlier. Meanwhile, our approach will adapt to the current concept until decreasing to the lowest prequential error, which follows the topic drifting periods. It is also suitable for Tweets_R containing some gradual topic drifts with different drifting periods. These data

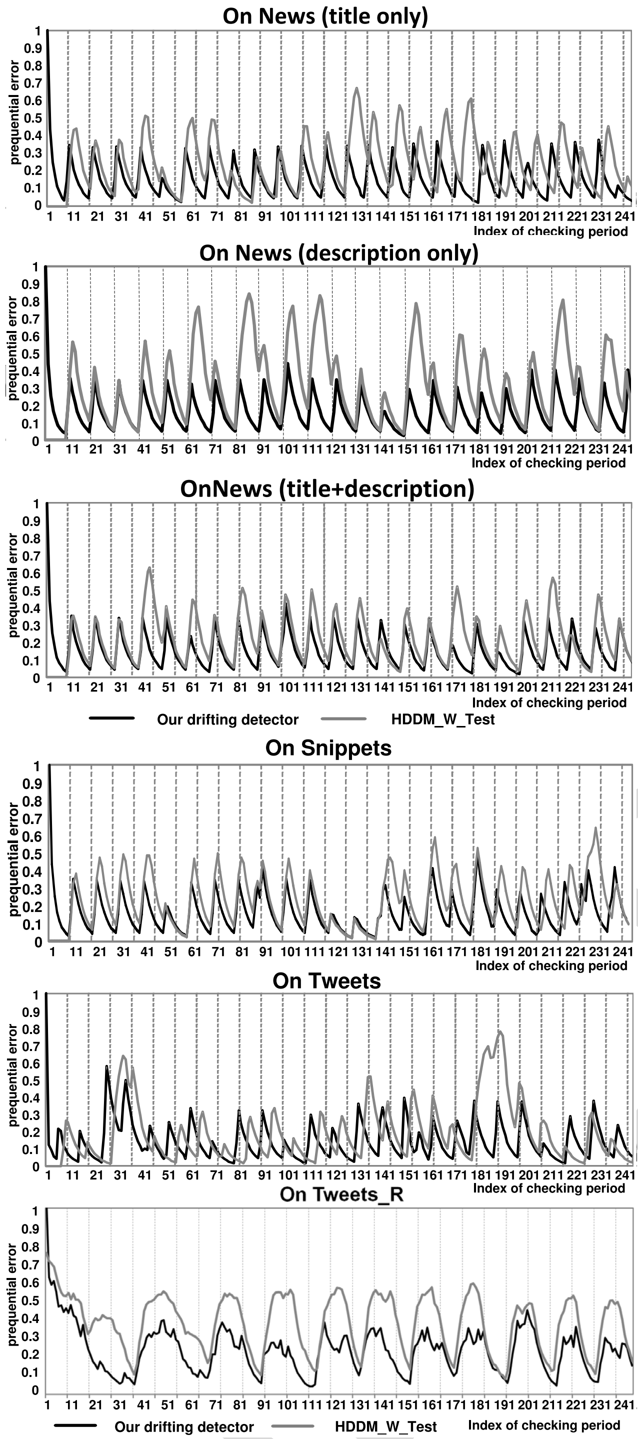


Fig. 6. Drift detection curves on data sets of news, snippets, and tweets.

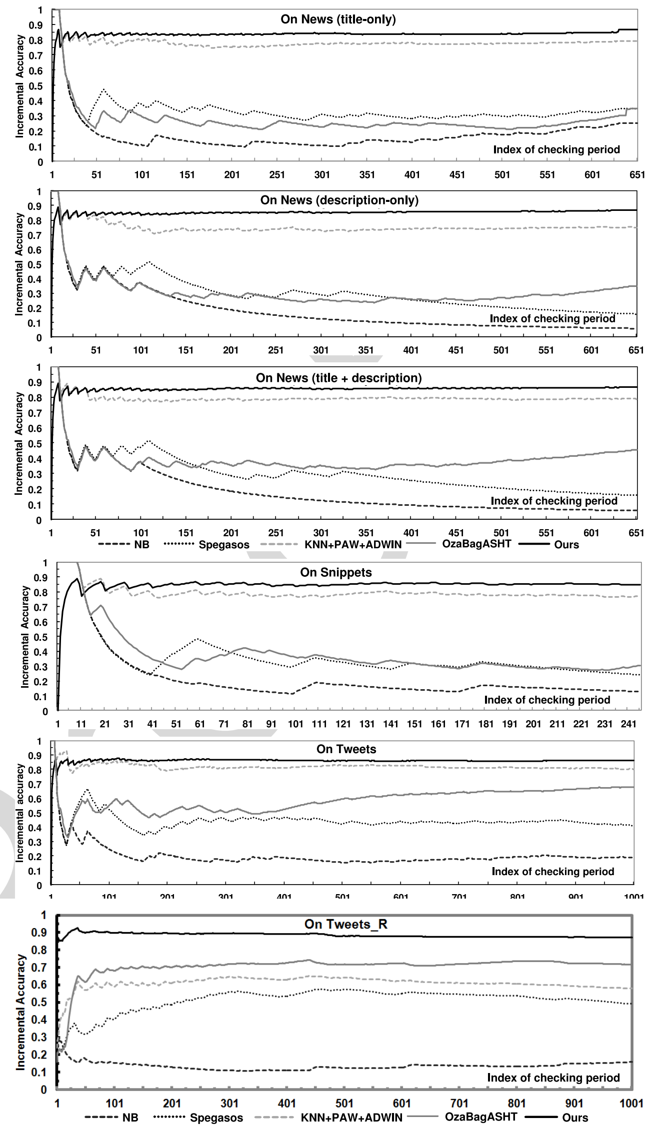


Fig. 7. Accuracy comparison between streaming classification approaches and ours on news, snippets, and tweets.

reveal that our concept cluster-based topic drifting detector can efficiently and effectively adapt to the topic drifts compared to the competing approach in synthetic and real short text streams with topic drifts and noisy data.

2) *Classification Performance:* We now want to investigate the classification performance in our approach compared to all competing ones. Fig. 5(b) shows the statistical test of all competing approaches over six data sets on the incremental accuracy. From this figure, we can see that on the incremental accuracy, our approach is the control algorithm, which is competitive to most of data stream approaches and the SVM/MaxEnt-based short text classification approaches. All of the above approaches are significantly superior to the data stream classification approach based on NB and the RF-based approach for short text classification. Reasons are analyzed below.

Regarding the NB-based approach, the accuracy of NB depends on the possibility statistics about both data distributions of classes and attributes, but all short text streams are high-dimensional and most of attributes are numerical, it is unbeneficial to get more informative probability statistics between classes and attributes. Regarding the RF-based approach, though the external topic information is extended into training documents, however, RF randomly selects a set of attributes from high-dimensional and numerical ones to do the split-tests, which causes the inaccuracy split information and the worse performance in a single random decision tree. It is similar to that in OzaBagASHT based on Hoeffding Trees. Regarding the Spegasos approach, it is refined from the SVM

TABLE V
DRIFTING DETECTION STATISTICS WITH RANKING AVERAGING OVER FOUR BASE CLASSIFIERS ON DATA SETS [FALARM: FALSE ALARM(%), MISSING(%)]

	DDM	EDDM	Cusum-DM	EWMA-ChartDM	ADWIN	OnePass-SamplerR	PageHin-kleyDM	HDDM_A_Test	HDDM_W_Test	Our
On Snippets										
FAlarm	0.53(2)	**0.41(1)**	6.23(6)	67.53(10)	65.16(9)	47.71(8)	20.83(7)	6.17(5)	2.50(4)	0.82(3)
Missing	47.50(6)	60.00(7)	61.67(8)	75.00(9)	28.33(3)	25.83(2)	92.50(10)	43.33(4)	44.17(5)	**10.00(1)**
Delay	188(7)	151(6)	150(5)	109(4)	68(2)	202(8)	**63(1)**	320(10)	234(9)	89(3)
On News (title-only)										
FAlarm	0.72(2)	**0.39(1)**	31.04(7)	71.85(10)	69.18(9)	56.83(8)	4.76(3)	11.46(6)	5.31(4)	10.24(5)
Missing	58.33(7)	56.02(6)	64.81(8)	79.17(9)	24.07(2)	28.24(3)	94.44(10)	43.52(4)	45.37(5)	**3.61(1)**
Delay	242(8)	238(7)	105(3)	145(5)	116(4)	191(6)	55(2)	348(10)	285(9)	**30(1)**
On News (description-only)										
FAlarm	0.33(2)	**0.20(1)**	16.75(7)	67.04(10)	54.55(9)	50.35(8)	2.78(5)	5.89(6)	1.71(3)	2.74(4)
Missing	66.37(5)	66.37(5)	68.75(8)	80.36(9)	52.68(3)	46.13(2)	94.35(10)	66.71(7)	63.39(4)	**23.21(1)**
Delay	180(6)	192(7)	156(4)	88(3)	56(1)	176(5)	200(8)	391(10)	377(9)	**56(1)**
On News (title+description)										
FAlarm	0.53(2)	**0.52(1)**	22.10(7)	52.40(8)	66.17(10)	52.49(9)	4.17(4)	12.08(6)	10.83(5)	0.77(3)
Missing	69.05(7)	66.67(6)	79.17(8)	80.95(9)	44.64(2)	45.54(3)	94.64(10)	56.85(5)	55.06(4)	**19.05(1)**
Delay	142(5)	162(6)	87(4)	75(2)	**68(1)**	168(7)	206(8)	337(9)	340(10)	86(3)
On Tweets										
FAlarm	**0.29(1)**	0.41(2)	15.20(6)	34.67(8)	67.48(10)	53.21(9)	16.36(7)	9.74(5)	3.77(3)	8.01(4)
Missing	61.19(7)	54.03(6)	72.09(8)	75.21(9)	**24.25(1)**	26.04(2)	96.61(10)	44.66(4)	45.27(5)	26.12(3)
Delay	245(8)	220(5)	252(10)	**97(1)**	173(3)	212(4)	235(6)	242(7)	245(8)	138(2)
On Tweets_R										
FAlarm	0.99(4)	0.98(2)	0.99(4)	1.00(10)	0.99(4)	0.99(4)	0.98(2)	0.99(4)	0.99(4)	**0.34(1)**
Missing	0.98(8)	0.86(3)	0.98(8)	1.00(10)	0.85(2)	0.93(4)	0.97(7)	0.93(4)	0.95(6)	**0.32(1)**
Delay	24(3)	25(4)	36(8)	/(10)	26(5)	49(9)	31(7)	18(2)	26(5)	**14(1)**

approach. Contrary to S_SVM using a sliding window, the model in Spegasos is incrementally built and updated with the arriving of each instance in the handling of a short text stream. Thus, Spegasos cannot perform as well as S_SVM compared to our approach, because of using insufficient statistical information of instances.

In the other dimension, we also investigate experimental results predicted by our approach and all competing ones. Fig. 7 first reports the curves of incremental accuracy predicted by our approach and four data stream classification approaches on six data sets. Regarding the final stable curves, we can draw a conclusion that our approach can beat all competing ones. The reason is analyzed below. We know that attribute values in a short text stream are very sparse, which causes the worse classification performance for all competing approaches. But the NB-based approach takes into account not only the prior distribution of the classes but the conditional probabilities of the attribute-values given the class. The impact in the classification accuracy is more than other competing ones, it hence performs worst. Considering the OzaBagASHT-based approach, it is an ensemble model based on Hoeffding trees and it requires the informed split-tests over attribute values. Though the ensemble model performs better than a single model, the data sparsity leads to the worse split-cuts. The corresponding prediction accuracy is also lower than that in Spegasos- and KNN+PAW+ADWIN-based approaches. Considering the Spegasos-based approach, it is refined from the SVM model, which can effectively tackle the sparse data. Thus, it is superior to the above two approaches. Considering the KNN+PAW+ADWIN-based approach, it also introduces the KNN mechanism as ours, the difference is that our approach uses the K nearest concept clusters instead of K nearest instances, more semantics are considered. Therefore, our approach outperforms all competing ones.

Furthermore, Fig. 8 reports the curves of incremental accuracy in our approach and four short text classification approaches. From the experimental results, we can get similar observations mentioned above. In addition, considering all Wikipedia resource-based approaches, all classifiers are trained over short texts extended from Wikipedia. The basic classifier RF is built on RF trees, but the training model built on the random selection of split-attributes will be seriously impacted from the data sparsity. Thus, it performs worst than other basic classifiers without informed split tests (e.g., SVM and MaxEnt). As compared with the SVM-based approaches, the MaxEnt-based approach is superior to others no matter whether the classifier learns from the extended short text or not. This is because MaxEnt is a framework for integrating information from many heterogeneous information sources for classification. It is robust and it is more suitable for classifying sparse data [18].

C. *Efficiency*

Fig. 5(b) shows the statistical test of all competing approaches over three data sets on the time consumption. Meanwhile, Fig. 9 compares the execution time between our approach and all competing ones in detail. According to the experimental results, we can observe the followings.

First, our approach is competitive to the control approach S_SVM. It is fastest due to using the linear classifier and the optimal method. However, the advantage of S_SVM

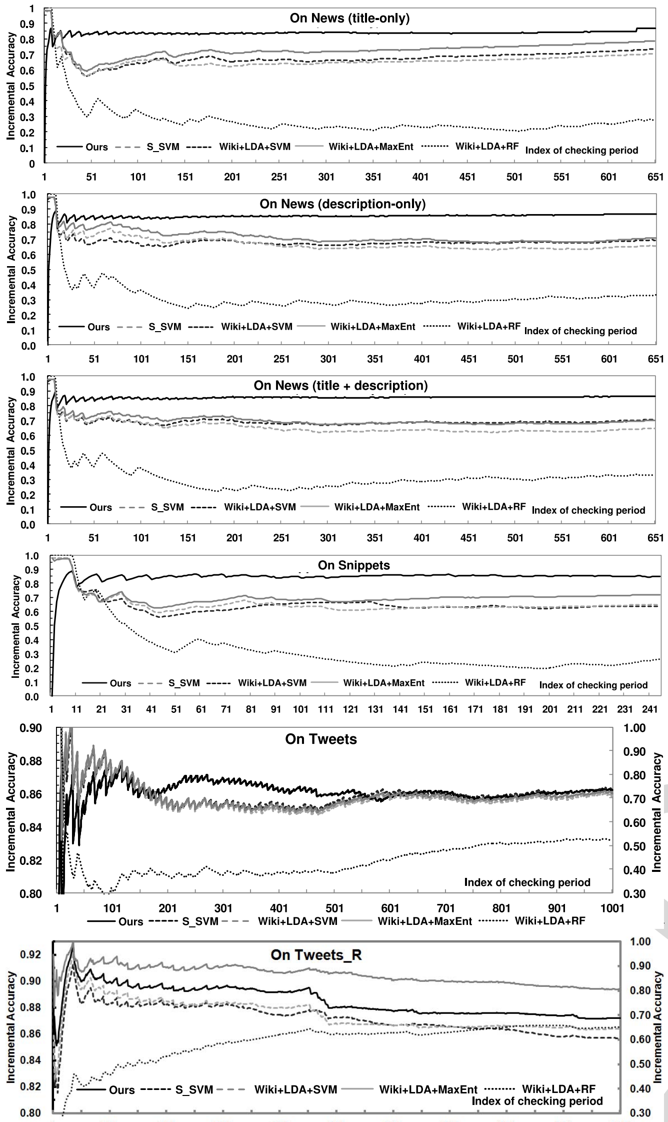


Fig. 8. Accuracy of short text classification approaches and ours on news, snippets, and tweets.

presents unobviously in the handling of large-scale Tweet data. Second, our approach costs lighter compared to all short text classification approaches using the topic information extracted from Wikipedia. This is because the acquisition of external resources and the topic modeling are very time-consuming. Third, as compared with data stream classification approaches, our approach is faster than OzaBagASHT- and KNN+PAW+ADWIN-based approaches. Because the time complexity of the OzaBagASHT-based approach is in direct proportion to the size of a sliding window (denoted as n') and the number of classifiers (denoted as C) and the iteration times (denoted as T), while the KNN+PAW+ADWIN-based approach is directly proportional to the square size of a sliding window. Due to usually $n' >> |D_i|$, $C > K$ and $T > |D_i|$, these two approaches also consume heavier time costs than ours. Meanwhile, as compared to NB- and Spegasos-based approaches, both time complexities are linear, while our approach requires generating concept clusters using refined k-means, the time complexity in the handling of each data

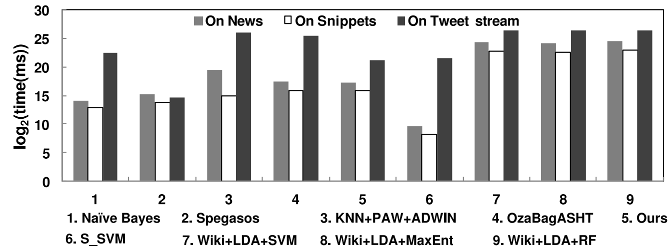


Fig. 9. Execution time of all competing approaches.

chunk with $|D_i|$ instances is up to $O(L|D_i|^2)$. If the training data is small, the former two approaches present the advantage, but they will be inferior to ours as the scale of the training data increases.

V. Conclusion

A new feature extension approach has been proposed for short text stream classification in this paper. Contrary to existing short text classification approaches, our proposed approach is first built on an incremental ensembling model to adapt to short text streams. Second, it uses an open semantic network Probase as the external resource to expand the feature space. That is, more semantic contexts based on the senses of terms hidden in short texts are introduced to make up of the data sparsity and all terms are disambiguated to reduce the noisy impact. Third, a topic drifting detection method is presented to track the topic changing. Finally, experimental results have revealed the effectiveness and the efficiency of our proposed approach. In our future work, larger scales of short text data sets will be investigated by collecting more real data sets.

References

[1] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Mach. Learn.*, vol. 39, nos. 2–3, pp. 103–134, 2000.

[2] X. Cheng, X. Yan, Y. Lan, and J. Guo, "BTM: Topic modeling over short texts," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 12, pp. 2928–2941, Dec. 2014.

[3] M. Chen, X. Jin, and D. Shen, "Short text classification improved by learning multi-granularity topics," in *Proc. IJCAI*, Barcelona, Spain, 2011, pp. 1776–1781.

[4] L. Gao, S. Zhou, and J. Guan, "Effectively classifying short texts by structured sparse representation with dictionary filtering," *Inf. Sci.*, vol. 323, pp. 130–142, Dec. 2015.

[5] X. Wang, Y. Wang, W. Zuo, and G. Cai, "Exploring social context for topic identification in short and noisy texts," in *Proc. AAAI*, Austin, TX, USA, 2015, pp. 1868–1874.

[6] D. Bollegala, Y. Matsuo, and M. Ishizuka, "A Web search engine-based approach to measure semantic similarity between words," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 7, pp. 977–990, Jul. 2011.

[7] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[8] M. Shirakawa, K. Nakayama, T. Hara, and S. Nishio, "Wikipedia-based semantic similarity measurements for noisy short texts using extended naive Bayes," *IEEE Trans. Emerg. Topics Comput.*, vol. 3, no. 2, pp. 205–219, Jun. 2015.

[9] W. Wu, H. Li, H. Wang, and K. Q. Zhu, "Probase: A probabilistic taxonomy for text understanding," in *Proc. SIGMOD*, Scottsdale, AZ, USA, 2012, pp. 481–492.

[10] Q. Yuan, G. Cong, and N. M. Thalmann, "Enhancing naive Bayes with various smoothing methods for short text classification," in *Proc. WWW*, Lyon, France, 2012, pp. 645–646.

[11] P. Wang *et al.*, "Semantic clustering and convolutional neural network for short text categorization," in *Proc. ACL*, Beijing, China, 2015, pp. 352–357.

[12] M. Haddoud, A. Mokhtari, T. Lecroq, and S. Abdeddaïm, "Combining supervised term-weighting metrics for SVM text classification with extended term representation," *Knowl. Inf. Syst.*, vol. 49, no. 3, pp. 909–931, 2016.

[13] P. Bicalho, M. Pita, G. Pedrosa, A. Lacerda, and G. L. Pappa, "A general framework to expand short text for topic modeling," *Inf. Sci.*, vol. 393, pp. 66–81, Jul. 2017.

[14] N. D. Doulamis, A. D. Doulamis, P. Kokkinos, and E. M. Varvarigos, "Event detection in Twitter microblogging," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2810–2824, Dec. 2016.

[15] X. Xu, C. Gao, J. Pei, K. Wang, and A. Al-Barakati, "Continuous similarity search for evolving queries," *Knowl. Inf. Syst.*, vol. 48, no. 3, pp. 649–678, 2016.

[16] Y. D. Zhai, K. P. Wang, D. N. Zhang, L. Huang, and C. Zhou, "An algorithm for semantic similarity of short text based on wordnet," *Acta Electronica Sinica*, vol. 40, no. 3, pp. 617–620, 2012.

[17] Z. Yu, H. Wang, X. Lin, and M. Wang, "Understanding short texts through semantic enrichment and hashing," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 2, pp. 566–579, Feb. 2016.

[18] X.-H. Phan *et al.*, "A hidden topic-based framework toward building applications with short Web documents," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 7, pp. 961–976, Jul. 2011.

[19] P. Wang, H. Zhang, Y.-F. Wu, B. Xu, and H.-W. Hao, "A robust framework for short text categorization based on topic model and integrated classifier," in *Proc. IJCNN*, Beijing, China, 2014, pp. 3534–3539.

[20] A. Bouaziz, C. Dartigues-Pallez, C. da Costa Pereira, F. Precioso, and P. Lloret, "Short text classification using semantic random forest," in *Proc. DaWaK*, Munich, Germany, 2014, pp. 288–299.

[21] Y. Zuo *et al.*, "Topic modeling of short texts: A pseudo-document view," in *Proc. KDD*, San Francisco, CA, USA, 2016, pp. 2105–2114.

[22] J. Xuan, J. Lu, G. Zhang, and X. Luo, "Topic model for graph mining," *IEEE Trans. Cybern.*, vol. 45, no. 12, pp. 2792–2803, Dec. 2015.

[23] Y. Zuo, J. Zhao, and K. Xu, "Word network topic model: A simple but general solution for short and imbalanced texts," *Knowl. Inf. Syst.*, vol. 48, no. 2, pp. 379–398, 2016.

[24] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in *Proc. COLING*, Nantes, France, 1992, pp. 539–545.

[25] P. Li, H. Wang, K. Q. Zhu, Z. Wang, and X. Wu, "Computing term similarity by large probabilistic ISA knowledge," in *Proc. PCIKM*, San Francisco, CA, USA, 2013, pp. 1401–1410.

[26] F. Wang, Z. Wang, Z. Li, and J.-R. Wen, "Concept-based short text classification and ranking," in *Proc. CIKM*, Shanghai, China, 2014, pp. 1069–1078.

[27] Z. Wang, L. Shou, K. Chen, G. Chen, and S. Mehrotra, "On summarization and timeline generation for evolutionary tweet streams," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1301–1315, May 2015.

[28] C. Chen, F. Li, B. C. Ooi, and S. Wu, "Ti: An efficient indexing mechanism for real-time search on tweets," in *Proc. SIGMOD*, Athens, Greece, 2011, pp. 649–660.

[29] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "Moa: Massive online analysis," *Mach. Learn. Res.*, vol. 11, pp. 1601–1604, Mar. 2010.

[30] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Proc. SBIA*, 2004, pp. 286–295.

[31] M. Baena-Garća *et al.*, "Early drift detection method," in *Proc. Workshop KDDS*, Berlin, Germany, 2006, pp. 77–86.

[32] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in *Proc. SIAM*, Minneapolis, MN, USA, 2007, pp. 443–448.

[33] M. Severo and J. Gama, "Change detection with Kalman filter and CUSUM," in *Proc. DS*, Barcelona, Spain, 2006, pp. 243–254.

[34] R. Pears, S. Sakthithasan, and Y. S. Koh, "Detecting concept change in dynamic data streams," *Mach. Learn.*, vol. 97, no. 3, pp. 259–293, 2014.

[35] A. Bifet, G. D. F. Morales, J. Read, G. Holmes, and B. Pfahringer, "Efficient online evaluation of big data stream classifiers," in *Proc. KDD*, Sydney, NSW, Australia, 2015, pp. 59–68.

[36] I. Frias-Blanco *et al.*, "Online and non-parametric drift detection methods based on hoeffding's bounds," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 3, pp. 810–823, Mar. 2015.

[37] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for SVM," in *Proc. ICML*, Corvallis, OR, USA, 2007, pp. 807–814.

[38] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà, "New ensemble methods for evolving data streams," in *Proc. KDD*, Paris, France, 2009, pp. 139–148.

[39] H.-F. Yu, C.-H. Ho, Y.-C. Juan, and C.-J. Lin, "LibShortText: A library for short-text classification and analysis," Dept. Comput. Sci., Nat. Taiwan Univ., Taipei, Taiwan, Tech. Rep., 2013. [Online]. Available: https://www.csie.ntu.edu.tw/∼cjlin/libshorttext/

[40] J. Gama, R. Sebastiâo, and P. P. Rodrigues, "On evaluating stream learning algorithms," *Mach. Learn. Res.*, vol. 90, no. 3, pp. 317–346, 2013.

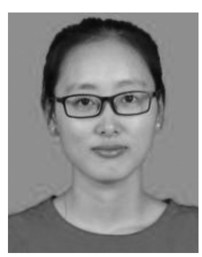
[41] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.

[42] Z. Yu *et al.*, "A new kind of nonparametric test for statistical comparison of multiple classifiers over multiple datasets," *IEEE Trans. Cybern.*, to be published.

Peipei Li received the B.S., M.S., and Ph.D. degrees from the Hefei University of Technology, Hefei, China, in 2005, 2008, and 2013, respectively.

She is currently an Associate Professor with the Hefei University of Technology. She was a Research Fellow with Singapore Management University, Singapore, from 2008 to 2009. She was a Student Intern with Microsoft Research Asia, Beijing, China, from 2011 to 2012. Her current research interests include data stream mining and knowledge engineering.

Lu He received the B.S. degree from Taishan Medical University, Taishan, China, in 2014. She is currently pursuing the postgraduate degree with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China. Her current research interest includes multilabel data stream classification.



Haiyan Wang received the B.S. degree from the Hefei University of Technology, Hefei, China, in 2016, where she is currently pursuing the postgraduate degree with the School of Computer Science and Information Engineering.

Her current research interest includes short text stream classification.

Xuegang Hu received the B.S. degree from the Department of Mathematics, Shandong University, Jinan, China, and the M.S. and Ph.D. degrees with the Hefei University of Technology, Hefei, China.

He is a Professor with the School of Computer Science and Information Engineering, Hefei University of Technology, and the Director-General with the Computer Association of Higher Education, Anhui Province, China. His current research interests include data mining and knowledge engineering.

Yuhong Zhang received the B.S., M.S., and Ph.D. degrees from the Hefei University of Technology, Hefei, China, in 2001, 2004, and 2011, respectively.

She is an Associate Professor with the School of Computer Science and Information Engineering, Hefei University of Technology. Her current research interests include transfer learning, data stream classification, and data mining.

Lei Li (SM'17) received the B.S. degree from Jilin University, Changchun, China, in 2004, the M.S. degree from the Memorial University of Newfoundland, St. John's, NL, USA, in 2006, and the Ph.D. degree from Macquarie University, Sydney, NSW, Australia, in 2012.

He is an Associate Professor with the Hefei University of Technology, Hefei, China. His current research interests include graph computing, social computing, and data mining.

Xindong Wu (F'11) received the Ph.D. degree in artificial intelligence from the University of Edinburgh, Edinburgh, U.K.

He is a Yangtze River Scholar with the Hefei University of Technology, Hefei, China, and a Professor of computer science with the University of Louisiana at Lafayette, Lafayette, LA, USA. His current research interests include data mining and big data analytics.

Dr. Wu is the Steering Committee Chair of ICDM and the Editor-in-Chief of Knowledge and Information Systems. He is a fellow of the AAAS.