Hierarchical Multi-Label Classification of Social Text Streams

Zhaochun Ren
University of Amsterdam
Amsterdam, The Netherlands
z.ren@uva.nl

Maria-Hendrike Peetz
University of Amsterdam
Amsterdam, The Netherlands
m.h.peetz@uva.nl

Shangsong Liang
University of Amsterdam
Amsterdam, The Netherlands
s.liang@uva.nl

Willemijn van Dolen
Business School
University of Amsterdam
Amsterdam, The Netherlands
w.m.vandolen@uva.nl

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
derijke@uva.nl

ABSTRACT

Hierarchical multi-label classification assigns a document to multiple hierarchical classes. In this paper we focus on hierarchical multi-label classification of social text streams. Concept drift, complicated relations among classes, and the limited length of documents in social text streams make this a challenging problem. Our approach includes three core ingredients: short document expansion, time-aware topic tracking, and chunk-based structural learning. We extend each short document in social text streams to a more comprehensive representation via state-of-the-art entity linking and sentence ranking strategies. From documents extended in this manner, we infer dynamic probabilistic distributions over topics by dividing topics into dynamic "global" topics and "local" topics. For the third and final phase we propose a chunk-based structural optimization strategy to classify each document into multiple classes. Extensive experiments conducted on a large real-world dataset show the effectiveness of our proposed method for hierarchical multi-label classification of social text streams.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering

Keywords

Twitter; tweet classification; topic modeling; structural SVM

1. INTRODUCTION

The growth in volume of social text streams, such as microblogs and web forum threads, has made it critical to develop methods that facilitate understanding of such streams. Recent work has confirmed that short text classification is an effective way of assisting users in understanding documents in social text streams [25, 26, 29, 46]. Straightforward text classification methods, however, are not adequate for mining documents in social streams.

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.

ACM 978-1-4503-2257-7/14/07 ... $15.00.
http://dx.doi.org/10.1145/2600428.2609561

For many social media applications, a document in a social text stream usually belongs to multiple labels that are organized in a hierarchy. This phenomenon is widespread in web forums, question answering platforms, and microblogs [11]. In Fig. 1 we show an example of several classes organized in a tree-structured hierarchy, of which several subtrees have been assigned to individual tweets. The tweet "I think the train will soon stop again because of snow ..." is annotated with multiple hierarchical labels: "Communication," "Personal experience" and "Complaint." Faced with many millions of documents every day, it is impossible to manually classify social streams into multiple hierarchical classes. This motivates the *hierarchical multi-label classification* (HMC) task for social text streams: classify a document from a social text stream using multiple labels that are organized in a hierarchy.

Recently, significant progress has been made on the HMC task, see, e.g., [4, 7, 10]. However, the task has not yet been examined in the setting of social text streams. Compared to HMC on stationary documents, HMC on documents in social text streams faces specific challenges: (1) Because of *concept drift* a document's statistical properties change over time, which makes the classification output different at different times. (2) The shortness of documents in social text streams hinders the classification process.

In this paper, we address the HMC problem for documents in social text streams. We utilize structural support vector machines (SVMs) [41]. Unlike with standard SVMs, the output of structural SVMs can be a complicated structure, e.g., a document summary, images, a parse tree, or movements in video [22, 45]. In our case, the output is a 0/1 labeled string representing the hierarchical classes, where a class is included in the result if it is labeled as 1. For example, the annotation of the top left tweet in Fig. 1 is `1100010000100`. Based on this structural learning framework, we use multiple structural classifiers to transform our HMC problem into a chunk-based classification problem. In chunk-based classification, the hierarchy of classes is divided into multiple chunks.

To address the shortness and concept drift challenges mentioned above, we proceed as follows. Previous solutions for working with short documents rely on extending short documents using a large external corpus [32]. In this paper, we employ an alternative strategy involving both entity linking [30] and sentence ranking to collect and filter relevant information from Wikipedia. To address concept drift [1, 39], we track dynamic statistical distributions of topics over time. Time-aware topic models, such as dynamic topic mod-

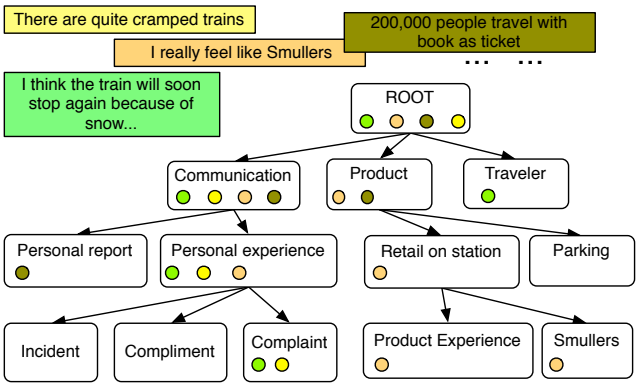


Figure 1: An example of predefined labels in hierarchical multi-label classification of documents in a social text stream. Documents are shown as colored rectangles, labels as rounded rectangles. Circles in the rounded rectangles indicate that the corresponding document has been assigned the label. Arrows indicate hierarchical structure between labels.

els (DTM) [5], are not new. Compared to latent Dirichlet allocation (LDA) [6], dynamic topic models are more sensitive to bursty topics. A *global* topic is a stationary latent topic extracted from the whole document set and a *local* topic is a dynamic latent topic extracted from a document set within a specific time period. To track dynamic topics, we propose an extension of DTM that extracts both global and local topics from documents in social text streams.

Previous work has used Twitter data for streaming short text classification [29]. So do we. We use a large real-world dataset of tweets related to a major public transportation system in a European country to evaluate the effectiveness of our proposed methods for hierarchical multi-label classification of documents in social text streams. The tweets were collected and annotated as part of their online reputation management campaign. As we will see, our proposed method offers statistically significant improvements over state-of-the-art methods.

Our contributions can be summarized as follows:

- We present the task of hierarchical multi-label classification for streaming short texts.
- We use document expansion to address the shortness issue in the HMC task for short documents, which enriches short texts using Wikipedia articles. We tackle the time-aware challenge by developing a new dynamic topic model that distinguishes between local topics and global topics.
- Based on a structural learning framework, we transform our hierarchical multi-label classification problem into a chunk-based classification problem via multiple structural classifiers, which is shown to be effective in our experiments using a large-scale real-world dataset.

We introduce related work in §2; in §3 we formulate our research problem. We describe our approach in §4; §5 details our experimental setup and §6 presents the results; §7 concludes the paper.

2. RELATED WORK

2.1 Short text classification

In recent years, short text classification has received considerable attention. Most previous work in the literature addresses the sparseness challenge by extending short texts using external knowledge. Those techniques can be classified into web search-based methods and topic-based ones.

Web search-based methods handle each short text as a query to a search engine, and then improve short text classification performance using external knowledge extracted from web search engine results [8, 44]. Such approaches face efficiency and scalability challenges, which makes them ill-suited for use in our data-rich setting [13]. As to topic-based techniques, Phan et al. [32] extract topic distributions from a Wikipedia dump based on the LDA [6] model. Similarly, Chen et al. [13] propose an optimized algorithm for extracting multiple granularities of latent topics from a large-scale external training set; see [37] for a similar method.

Besides those two strategies, other methods have also been employed. E.g., Nishida et al. [28], Sun [38] improve classification performance by compressing shorts text into entities. Zhang et al. [46] learn a short text classifier by connecting what they call the "information path," which exploits the fact that some instances of test documents are likely to share common discriminative terms with the training set. Few previous publications on short text classification consider a streaming setting; none focuses on a hierarchical multiple-label version of the short text classification problem.

2.2 Hierarchical multi-label classification

In the machine learning field, *multi-label classification* problems have received lots of attention. Discriminative ranking methods have been proposed in [36], while label-dependencies are applied to optimize the classification results by [18, 20, 31]. However, none of them can work when labels are organized hierarchically.

The *hierarchical* multi-label classification problem is to classify a given document into multiple labels that are organized as a hierarchy. Koller and Sahami [19] propose a method using Bayesian classifiers to distinguish labels; a similar approach uses a Bayesian network to infer the posterior distributions over labels after training multiple classifiers [3]. As a more direct approach to the HMC task, Rousu et al. [34] propose a large margin method, where a dynamic programming algorithm is applied to calculate the maximum structural margin for output classes. Decision-tree based optimization has also been applied to the HMC task [7, 42]. Cesa-Bianchi et al. [10] develop a classification method using hierarchical SVM, where SVM learning is applied to a node if and only if this node's parent has been labeled as positive. Bi and Kwok [4] reformulate the "tree-" and "DAG-" hierarchical multi-label classification tasks as problems of finding the best subgraph in a tree and DAG structure, by developing an approach based on kernel density estimation and the condensing sort and select algorithm.

To the best of our knowledge there is no previous work on HMC for (short) documents in social text streams. Additionally, we present a chunk-based structural learning method for the HMC task, which is different from existing HMC approaches, and which we show to be effective for the traditional stationary case and the streaming case.

3. PRELIMINARIES

We detail the task that we address and introduce important concepts, including preliminaries about structural SVMs.

3.1 Problem formulation

We begin by defining the hierarchical multi-label classification (HMC) task. We are given a class hierarchy $(C, \prec)$, where C is a set of class labels and $\prec$ is a partial order representing the parent relationship, i.e., $\forall c_i, c_j \in C$, $c_i \prec c_j$ if and only if c_i is the parent class of c_j. We write $\mathbf{x}^{(i)}$ to denote a feature vector, i.e., an element of the feature space $\mathcal{X}$, and we write $\mathbf{y}^{(i)} \in \{0,1\}^{|C|}$

for the target labeling. Let D be the set of input documents, and $|D|$ the size of D. The target of a hierarchical multi-label classifier, whether for stationary documents or for a stream of documents, is to learn a hypothesis function $f\colon \mathcal{X} \to \{0,1\}^C$ from training data $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{|D|}$ to predict a $\mathbf{y}$ when given $\mathbf{x}$. Suppose the hierarchy is a tree structure. Then, classes labeled positive by $\mathbf{y}$ must satisfy the $\mathcal{T}$-*property* [4]: if a labeled $c \in C$ is labeled positive in output $\mathbf{y}$, its parent label must also be labeled positive in $\mathbf{y}$. Given the $\mathcal{T}$-*property*, we define a *root* class r in the beginning of each C, which refers to the root vertex in HMC tree structure. Thus for each $\mathbf{y}$ in HMC, we have $\mathbf{y}^{(r)} = 1$.

Hierarchical multi-label classification for short documents in social streams (HMC-SST) learns from previous time periods and predicts an output when a new document arrives. More precisely, given a class hierarchy $(C, \prec)$ and a collection of documents seen so far, $\mathcal{X} = \{X_1, \ldots, X_{t-1}\}$, HMC-SST learns a hypothesis function $f\colon \mathcal{X} \to \{0,1\}^C$ that evolves over time. Thus, at time period $t, t > 1$, we are given a function f that has been trained during the past $t-1$ periods and a set of newly arriving documents X_t. For each $\mathbf{x}_t^{(i)} \in X_t$, $f(x)$ predicts $\hat{y}_t^{(i)}$ that labels each class $c \in C$ as 0 or 1. Classes in C that are labeled positive must follow the $\mathcal{T}$-property. Afterwards, f updates its parameters using X_t and their true labels $\{\mathbf{y}_t^{(i)}\}_{i=1}^{|X_t|}$.

Concept drift indicates the phenomenon that topic distributions change between adjacent time periods [17]. In streaming classification of documents [29] this problem needs to be addressed. We assume that each document in a stream of documents is concerned with multiple topics. By dividing the timeline into time periods, we dynamically track latent topics to cater the phenomenon of *concept drift* over time. For streaming documents, global statistics such as tf-idf or topic distributions cannot reflect drift phenomena. However, local statistics derived from a specific period are usually helpful for solving this problem [5, 21, 29]. Ideally, one would find a trade-off between tracking the extreme local statistics and extreme global statistics [21]. Thus, in this paper we address the issue of concept drift by tracking both global topics (capturing the complete corpus) and local, latent and temporally bounded, topics over time. Given a document set X_t published at time t, we split the topic set Z_t into $Z_t^g \cup Z_t^l$, with global topics Z_t^g that depend on all time periods and documents seen so far, and local topics Z_t^l derived from the previous period $t-1$ only. We then train our temporal classifier incrementally based on those global and local topic distributions.

3.2 Structural SVMs

Structural SVMs have been proposed for complex classification problems in machine learning [22, 23, 35]. We follow the notation from [41]. Given an input instance $\mathbf{x}$, the target is to predict the structured label $\mathbf{y}$ from the output space $\mathcal{Y}$ by maximizing a discriminant $\mathcal{F} : \mathcal{X} \times \mathcal{Y} \to \Re$:

$$\mathbf{y} = f(\mathbf{x}; \mathbf{w}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} \mathcal{F}(\mathbf{x}, \mathbf{y}; \mathbf{w}), \quad (1)$$

where the discriminant $\mathcal{F}$ measures the correlation between $(\mathbf{x}, \mathbf{y})$, and $\mathbf{w}$ indicates the weights of $\mathbf{x}$ in $\mathcal{F}$. The discriminant $\mathcal{F}$ will get its maximal value when $\mathbf{y} = f(\mathbf{x}; \mathbf{w})$, which is set as hypothesis function in HMC-SST. We assume the discriminant $\mathcal{F}$ to be linear in a joint feature space $\Psi : X \times Y \to R^K$, thus $\mathcal{F}$ can be rewritten as $\mathcal{F}(x, y; w) = \langle \mathbf{w}, \Psi(\mathbf{x}, \mathbf{y}) \rangle$. The feature mapping Ψ maps the pair $(\mathbf{x}, \mathbf{y})$ into a suitable feature space endowed with the dot product. Then the function $\mathcal{F}$ can be learned in a large-margin framework through the training set $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{T}$ by minimizing the objective function:

$$\min_{\zeta \geq 0} \tfrac{1}{2}\|w\|^2 + C \textstyle\sum_{i=1}^{n} \zeta_i \quad (2)$$

such that for all i and all $\mathbf{y} \in Y \backslash \mathbf{y}^{(i)}$:

$$w^T \Psi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) - w^T \Psi(\mathbf{x}^{(i)}, \mathbf{y}) \geq \Delta(\mathbf{y}, \mathbf{y}^{(i)}) - \zeta_i, \quad (3)$$

where $w^T \Psi(\mathbf{x}^{(i)}, \mathbf{y})$ indicates the hypothesis function value given $\mathbf{x}^{(i)}$ and a random $\mathbf{y}$ from $Y \backslash \mathbf{y}^{(i)}$. For each $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$, a set of constraints (see Eq. 3) is added to optimize the parameters w. Note that $\mathbf{y}^{(i)}$ is the prediction that minimizes the loss function $\Delta(\mathbf{y}, \mathbf{y}^{(i)})$. The loss function equals 0 if and only if $\mathbf{y} = \mathbf{y}^{(i)}$, and it decreases when $\mathbf{y}$ and $\mathbf{y}^{(i)}$ become more similar. Given the exponential size of Y, the number of constraints in Eq. 3 makes the optimization challenging.

4. METHOD

We start by providing an overview of our approach to HMC for documents in social text streams. We then detail each of our three main steps: document expansion, topic modeling and incremental structural SVM learning.

4.1 Overview

We provide a general overview of our scenario for performing HMC on (short) documents in social text streams in Fig. 2. There are three main phases: (A) document expansion; (B) time-aware topic modeling; (C) chunk-based structural classification. To summarize, at time period t_i, we are given a temporally ordered short documents set $X_{t_i} = \{\mathbf{x}_{t_i}^{(1)}, \mathbf{x}_{t_i}^{(2)}, \ldots, \mathbf{x}_{t_i}^{(|X_t|)}\}$. For each short text $\mathbf{x}_{t_i} \in X_{t_i}$, in phase (A) (see §4.2) we expand $\mathbf{x}_{t_i}$ through entity linking and query-based sentence ranking; we obtain $\mathbf{x}'_{t_i}$ from $\mathbf{x}_{t_i}$ by extracting relevant sentences from related Wikipedia articles.

Next, in phase (B) (see §4.3), we extract dynamic topics Φ_{t_i}; building on an extended DTM model, we extract both global and local topical distributions for $\mathbf{x}'_{t_i}$; then, a feature vector for $\mathbf{x}'_{t_i}$ is generated as $\Psi(\mathbf{x}'^{(i)}, \mathbf{y})$.

Based on the extracted features, we train an incremental chunk-based structural learning framework in (C) in §4.4. We introduce multiple structural classifiers to the optimization problem by transferring the set of classes C to another representation using multiple chunks $\mathcal{S}$. Traversing from the most abstract chunk $r_{\mathcal{S}} \in \mathcal{S}$, we define each chunk $s \in \mathcal{S}$ to be a set of chunks or classes. Leaves in $\mathcal{S}$ only include classes. For each chunk $sc \in \mathcal{S}$, we employ a discriminant to address the optimization problem over parameters $\mathcal{F}_{sc}$, where sc's child chunk/class will not be addressed unless it is labeled positive during our prediction. Accordingly, multiple discriminants are applied to predict labels given $\mathbf{x}_{t_i}$ and update their parameters based on true labels $\mathbf{y}_{t_i}$.

4.2 (A) Document expansion

To address the challenge offered by short documents, we propose a document expansion method that consists of two parts: entity linking and query-based sentence ranking and extraction.

4.2.1 *Entity linking*

Given a short document $\mathbf{x}_t$ at time t, the target of entity linking is to identify the entity e from a knowledge base E that is the most likely referent of $\mathbf{x}_t$. For each $\mathbf{x}_t$, a *link candidate* $e_i \in E$ links an *anchor* a in $\mathbf{x}_t$ to a target w, where an anchor is a word n-gram tokens in a document and each w is a Wikipedia article. A target is identified by its unique title in Wikipedia.

As the first step of our entity linking, we aim to identify as many *link candidates* as possible. We perform lexical matching of each n-gram anchor a of document d_t with the target texts found in Wikipedia, resulting in a set of *link candidates* E for each document d_t. As the second step, we employ the commonness (*CMNS*)

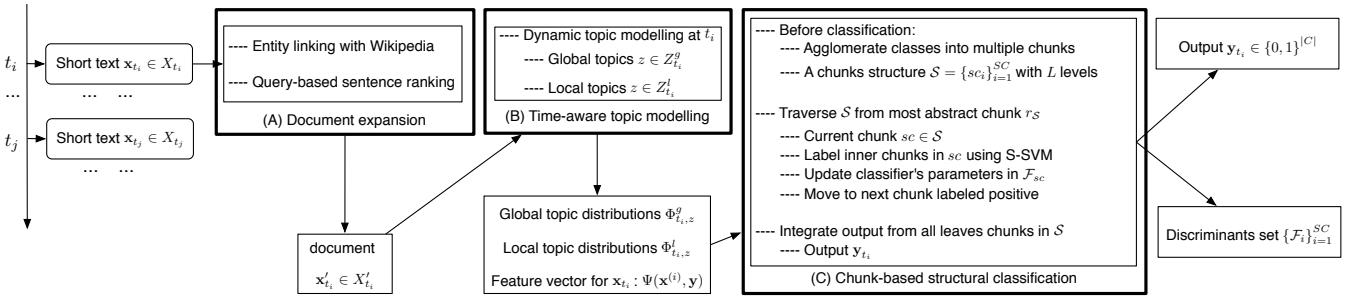


Figure 2: Overview of our approach to hierarchical multi-label classification of documents in social text streams. (A) indicates document expansion; (B) indicates the topic modeling process; (C) refers to chunk-based structural learning and classification.

method from [27] and rank link candidates E by considering the prior probability that anchor text a links to Wikipedia article w:

$$CMNS(a,w) = \frac{|E_{a,w}|}{\sum_{w' \in W} |E_{a,w'}|},$$

where $E_{a,w}$ is the set of all links with anchor text a and target w. The intuition is that link candidates with anchors that always link to the same target are more likely to be a correct representation. In the third step, we utilize a learning to rerank strategy to enhance the precision of correct link candidates. We extract a set of 29 features proposed in [27, 30], and use a decision tree-based approach to rerank the link candidates.

4.2.2 *Query-based sentence ranking*

Given the *link candidates* list, we extract the most central sentences from the top three most likely Wikipedia articles. As in LexRank [15], Markov random walks are employed to optimize the ranking list iteratively, where each sentence's score is voted from other sentences. First, we build the similarity matrix M, where each item in M indicates the similarity between two sentences given $\mathbf{x}_t$ as a query. Given two sentences s_i and s_j, we have:

$$M_{i,j} = sim(s_i, s_j|\mathbf{x}_t) / \sum_{j' \in |S|} sim(s_i, s_{j'}|\mathbf{x}_t) \quad (4)$$

At the beginning of the iterative process, an initial score for each sentence is set as $1/|S|$, and at the t-th iteration, the score of s_i is calculated as follows:

$$score(s_i)^{(t)} = (1-\lambda) \sum_{i \neq j} M_{i,j} \cdot score(s_j)^{(t-1)} + \lambda \frac{1}{|S|}, \quad (5)$$

where $|S|$ equals the number of sentences in Wikipedia documents that have been linked to the anchor text a in §4.2.1 and the damping factor $\lambda = 0.15$. Then the transition matrix $\widetilde{M}$ equals to:

$$\widetilde{M} = (1-\lambda)M + \bar{e}\bar{e}^T \lambda / |S|, \quad (6)$$

where $\bar{e}$ is a column vector with all items equal to 1. The iterative process will stop when it convergences. Since $\widetilde{M}$ is a column stochastic matrix, it can be proven that the value of *score* converges [43], and a value of *score* can be derived from the principle eigenvector of $\widetilde{M}$. We extract the top $\mathcal{E}_{\mathbf{x}_t}$ sentences from the ranked list, and extend $\mathbf{x}_t$ to $\mathbf{x}'_t$ by including those $\mathcal{E}_{\mathbf{x}_t}$ sentences in $\mathbf{x}_t$.

4.3 (B) Time-aware topic modeling

Concept drift makes tracking the change of topic distributions crucial for HMC of social text streams. We assume that each document in a social text stream can be represented as a probabilistic distribution over topics, where each topic is represented as a probabilistic distribution over words. The topics are not necessarily assumed to be stationary. We employ a dynamic extension of the LDA model to track latent dynamic topics. Comparing to previous work on dynamic topic models [5], our method is based on the conjugate prior between Dirichlet distribution and Multinomial distribution. To keep both stationary statistics and temporary statistics, we present a trade-off strategy between stationary topic tracking and dynamic topic tracking, where topic distributions evolve over time.

Fig. 3 shows our graphical model representation, where shaded and unshaded nodes indicate observed and latent variables, respectively. Among the variables related to document set X_t in the graph, z, θ, r are random variables and w is the observed variable; $|X_{t-1}|$, $|X_t|$ and $|X_{t+1}|$ indicate the number of variables in the model. As usual, directed arrows in a graphical model indicate the dependency between two variables; the variables ϕ^l_t depend on variables ϕ^l_{t-1}.

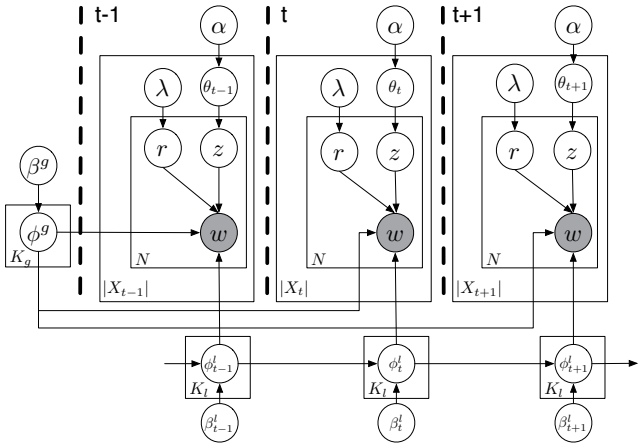


Figure 3: Graphical representation of topical modelling, where $t-1$, t and $t+1$ indicate three time periods.

The topic distributions $\theta_{\mathbf{x}_t}$ for a document $\mathbf{x}_t \in X_t$ are derived from a Dirichlet distribution over hyper parameter α. Given a word $w_i \in \mathbf{x}_t$, a topic z_{w_i} for word w_i is derived from a multinomial distribution $\theta_{\mathbf{x}_t}$ over document $\mathbf{x}_t$. We derive a probabilistic distribution ϕ_t over topics $Z_t = Z^g_t \cup Z^l_t$ from a Dirichlet distribution over hyper parameters b_t: if topic $z \in Z^l$, then $b_t = \beta^l_t \cdot \phi_{w_i,t-1}$, otherwise $b_t = \beta^g$. The generative process for our topic model at time $t > 1$, is described in Fig. 3.

Due to the unknown relation between ϕ_t and θ_t, the posterior distribution for each short text $\mathbf{x}_t$ is intractable. We apply Gibbs collapsed sampling [24] to infer the posterior distributions over both,

1. For each topic $z, z \in Z_t^l \cup Z_t^g$:

- Draw $\phi^g \sim Dirichlet(\beta^g)$;
- Draw $\phi_t^l \sim Dirichlet(\beta_t^l \cdot \phi_{t-1}^l)$;

2. For each candidate short text $\mathbf{x}_t \in X_t$:

- Draw $\theta_t \sim Dirichlet(\alpha_t)$;
- For each word w in d_t
  - Draw $r \sim Bernoulli(\lambda)$;
  - Draw $z_w \sim Multinomial(\theta_t)$;
    - if $r = 0$: Draw $w \sim Multinomial(\phi_z^g)$;
    - if $r = 1$: Draw $w \sim Multinomial(\phi_{z,t}^l)$;

Figure 4: Generative process for the topic model.

global and local topics. For each iteration during our sampling process, we derive the topic z via the following probability:

$$p(r_i = m, z_i = z|\mathcal{W}, Z_{-i}, \alpha, b_t) \propto \frac{n_{d,m,-i}^t + \lambda}{n_{d,-i}^t + 2\lambda} \cdot \frac{n_{d,z,-i}^t + \alpha}{\sum_{z' \in Z^m}(n_{d,z',-i}^t + \alpha)} \cdot \frac{n_{w,z,-i}^t + b_{w,z,t}^m}{\sum_{w' \in N_{u,t}} n_{w',z,-i}^t + N_t b_{w,z,t}^m}, \tag{7}$$

where m indicates the possible values of variable r for the ith word in document d_t, and the value m indicates the corresponding kind of topics when $r_i = m$. We set $b_{w,z,t} = \beta_t^l \cdot \phi_{w,z,t-1}$ when $r_i = 1$, and $b_{w,z,t} = \beta^g$ when $r_i = 0$. After sampling the probability for each topic z, we infer the posterior distributions for random variable $\phi_{w,z,t}$, which are shown as follows:

$$\begin{aligned} \phi_{w,z,t}^{r=0} &= \frac{n_{w,z,t} + \beta^g}{\sum_{z \in Z^m} n_{w,z,t} + \beta^g} \\ \phi_{w,z,t}^{r=1} &= \frac{n_{w,z,t} + \beta_t^l \cdot \phi_{w,z,t-1}}{\sum_{z \in Z^m} n_{w,z,t} + \beta_t^l \cdot \phi_{w,z,t-1}} \end{aligned} \tag{8}$$

4.4 (C) Chunk-based structural classification

Some class labels, specifically for some leaves of the hierarchy, only have very few positive instances. This skewedness is a common problem in hierarchical multi-label classification. To handle skewedness, we introduce a multi-layer chunk structure to replace the original class tree. We generate this chunk structure by employing a continuous agglomerative clustering approach to merge multiple classes/chunks to a more abstract chunk that contains a predefined number of items. Merging from classes, considered as leave nodes in the final chunk structure, our clustering strategy continues until what we call the *root chunk*, the most abstract chunk, has been generated. Following this process, we agglomerate the set of classes C into another set of chunks $\mathcal{S}$, each of which, denoted as sc, includes s items. During this continuous agglomerative clustering process from classes C to the *root chunk*, we define *successive* relations among chunks in $\mathcal{S}$. Each chunk sc's successive chunks/-classes in $\mathcal{S}$ are chunks/classes that exist as items in sc, i.e., chunk sc is a successive chunk of chunk sc^{pa} *iff* there exist a vertex in sc^{pa} corresponding to chunk sc.

Thus we can think of $\mathcal{S}$ as a tree structure. From the most abstract chunk $r_{\mathcal{S}} \in \mathcal{S}$ that is not included in any other chunk, each layer l of $\mathcal{S}$ is the set of child nodes in those chunks that exist in l's last layer. The leaves of $\mathcal{S}$ indicate classes. Then, a structural SVM classifier $\mathcal{F}_{sc}$ for chunk sc includes L_{sc} chunks, and its output space $\mathcal{Y}_{sc}$ refers to a set of binary labels $\{0, 1\}^{L_{sc}}$ over chunks.

At each time period t, we divide the HMC for documents in social text streams into a learning process and a inference process, which we detail below.

4.4.1 Learning with structural SVMs

For the learning process, we train multiple structural SVM classifiers from $\mathcal{S}$'s root chunk $r_{\mathcal{S}}$ to the bottom, where the $\mathcal{T}$-property must be followed by each chunk $sc \in \mathcal{S}$. After generating the chunk structure $\mathcal{S}$, we suppose $\mathcal{S}$ has SC chunks with L levels. At time t, we are given a set of training instances $\mathcal{T}_t = \{(\mathbf{x}_t^{(1)}, \mathbf{y}_t^{(1)}), (\mathbf{x}_t^{(2)}, \mathbf{y}_t^{(2)}), \ldots, (\mathbf{x}_t^{(|X_t|)}, \mathbf{y}_t^{(|X_t|)})\}$, and our target is to update parameters of multiple structural SVM classifiers during the learning process. Thus $\mathbf{y}_t^{(i)}$ in $(\mathbf{x}_t^{(i)}, \mathbf{y}_t^{(i)})$ is divided and extended into SC parts $\bigcup_{sc \in S}\{\mathbf{y}_{t,sc}^{(i)}\}$, where $\mathbf{y}_{t,sc}^{(i)}$ indicates the output vector in chunk sc. The structural classifier $\mathcal{F}_{sc}$ for chunk $sc \in \mathcal{S}, sc \neq r_c$, learns and updates its parameters after its parent chunk $p(sc)$ has received a positive label on the item corresponding to sc. For each chunk $sc \in \mathcal{S}$, we utilize the following structural SVM formulation to learn a weight vector $\mathbf{w}$, shown in Equation 9:

$$\min_{\zeta \geq 0} \frac{1}{2}\|\mathbf{w}_{t,sc}\|^2 + C\sum_{i=1}^{n}\zeta_i \tag{9}$$

subject to:

1. $\forall \mathbf{y}_{t,sc} \in \mathcal{Y}_{sc} \backslash \mathbf{y}_{t,sc}^{(i)}$;
2. $\forall c \in \mathbf{c}_{\mathbf{y}_{t,sc}}, p(c) \in \mathbf{c}_{\mathbf{y}_{t,sc}}$;
3. $w^T\Psi(\mathbf{x}_t^{(i)}, \mathbf{y}_{t,sc}^{(i)}) - w^T\Psi(\mathbf{x}^{(i)}, \mathbf{y}_{t,sc}) \geq \Delta(\mathbf{y}, \mathbf{y}_{t,sc}^{(i)}) - \zeta_i$;

where $\mathbf{c}_{\mathbf{y}_{t,sc}}$ are positive chunks labeled by $\mathbf{y}_{t,sc}^{(i)}$, and $\Psi(\mathbf{x}_t^{(i)}, \mathbf{y}_{t,sc})$ indicates the feature representation for $\mathbf{x}_t^{(i)}, \mathbf{y}_{t,sc}^{(i)}$.

Traditional SVMs only consider zero-one loss as a constraint during learning. This is inappropriate for complicated classification problems such as hierarchical multi-label classification. We define a loss function between two structured labels $\mathbf{y}$ and $\mathbf{y}_i$ based on their similarity as $\Delta(\mathbf{y}_{sc}, \mathbf{y}_{i,sc}) = 1 - sim(\mathbf{y}_{sc}, \mathbf{y}_{i,sc})$. Here, $sim(\mathbf{y}_{sc}, \mathbf{y}_{i,sc})$ indicates the structural similarity between two different subsets of sc's child sets $\mathbf{c}_{\mathbf{y}}$ and $\mathbf{c}_{\mathbf{y}^{(i)}}$. We compute the similarity between $\mathbf{y}_{t,sc}$ and $\mathbf{y}_{t,sc}^{(i)}$ by comparing the overlap of nodes in these two tree structures, as follows:

$$sim(\mathbf{y}_{t,sc}^{(i)}, \mathbf{y}_{t,sc}) = \frac{\sum_{n \in \mathbf{c}_{\mathbf{y}^{(i)}}, n' \in \mathbf{c}_{\mathbf{y}}} w_{n,n'} \cdot |(n \cap n')|}{\sum_{n \in \mathbf{c}_{\mathbf{y}^{(i)}}, n' \in \mathbf{c}_{\mathbf{y}}} w_{n,n'} \cdot |(n \cup n')|}, \tag{10}$$

where we set $w_{n,n'}$ to be the weight between two chunks n and n', each of which is included in $\mathbf{c}_{\mathbf{y}^{(i)}}$ and $\mathbf{c}_{\mathbf{y}}$ respectively. Since it is intractable to compare two chunks that are not at the same level in $\mathcal{S}$, here we set $w_{n,n'}$ to be:

$$w_{n,n'} = \begin{cases} 1/h_n & h_n = h_{n'} \\ 0 & else \end{cases} \tag{11}$$

To optimize Eq. 9, we adjust the cutting plane algorithm [16, 45] to maintain the $\mathcal{T}$-property. In general, the cutting plane algorithm iteratively adds constraints until the problem is solved by a desired tolerance ε. It starts with an empty set $\mathbf{y}_i$, for $i = 1, 2, \ldots, n$, and iteratively looks for the most violated constraint for $(\mathbf{x}_t^{(i)}, \mathbf{y}_{t,sc}^{(i)})$.

Algorithm 1: Cutting Plane Optimization for Equation 9

Input: $(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{x}^{(2)}, \mathbf{y}^{(2)}), ..., (\mathbf{x}^{(t)}, \mathbf{y}^{(t)}), C, \zeta$
$\mathbf{y}_i = \emptyset$;
repeat
  for $i = 1, 2, ... , n$ **do**
    $\omega \equiv w^T \Psi(x^{(i)}, y^{(i)}) - w^T \Psi(x^{(i)}, y)$;
    $H(y; w) \equiv \Delta(y^{(i)}, y) + \omega$;
    compute $\hat{y} = \arg\max_{\mathbf{y} \in Y} H(\mathbf{y}; \mathbf{w})$;
    repeat
      for *leaves node* $n \in sc$ **do**
        if $p(n) \notin \boldsymbol{c}_{\hat{y}}$ **then**
          $\hat{y}+ = \hat{y} \cup p(n)$;
          $\hat{y}- = \hat{y} - n$;
          $\hat{y} = \arg\max_y (H(\hat{y}+; w), H(\hat{y}-; w))$
        end
      end
    until $\hat{y} \in Y$ *hold* $\mathcal{T}$*-property*;
    if $H(\hat{y}; w) > \zeta_i + \varepsilon$ **then**
      $\mathbf{w} \leftarrow$ optimize Equation 9 over $\bigcup_i \{\mathbf{y}_i\}$
    end
  end
until *no working set has changed during iteration*;

Algorithm 1 shows that to maintain the $\mathcal{T}$-property, we adjust the set of positive chunks in $\hat{y}$ iteratively. The parameter $\mathbf{w}_{t,sc}$ is updated with respect to the combined working set $\bigcup_i \{\mathbf{y}_i\}$.

4.4.2 *Making predictions*

The feature representation for $\Psi(\mathbf{x}_t^{(i)}, \mathbf{y}_{t,sc})$ must enable meaningful discrimination between high quality and low quality predictions [45]. Our topic model generates a set of topical distributions, Φ_t, where each item $\phi(w|z,t) \in \Phi_t$ is a conditional distribution $P(w|z,t)$ over words w given topic z. Assuming that each document's saliency is summed up by votes from all words in the document, we then define $\Psi(\mathbf{x}, \mathbf{y})$ as follows:

$$\Psi(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} \frac{1}{N_\mathbf{x}} \sum_{w \in \mathbf{x}} \phi(w|z_1, t) \cdot \frac{1}{N_\mathbf{y}} n_{w,\mathbf{y}} \\ \frac{1}{N_\mathbf{x}} \sum_{w \in \mathbf{x}} \phi(w|z_2, t) \cdot \frac{1}{N_\mathbf{y}} n_{w,\mathbf{y}} \\ \vdots \\ \frac{1}{N_\mathbf{x}} \sum_{w \in \mathbf{x}} \phi(w|z_K, t) \cdot \frac{1}{N_\mathbf{y}} n_{w,\mathbf{y}} \end{bmatrix}, \quad (12)$$

where $n_{w,\mathbf{y}}$ indicates the number of times word w exist in $\mathbf{y}$ for the past $t-1$ periods; $N_\mathbf{x}$ refers to the number of words in documents $\mathbf{x}$ whereas $N_\mathbf{y}$ is the number of words in $\mathbf{y}$.

Given multiple structural SVMs $\mathcal{F}_{t,sc}$ that have been updated at time $t-1$, the target of our prediction is to select $\mathbf{y}_{t,sc}$ for instance $\mathbf{x}_t$ from the root chunk $r_\mathcal{S} \in \mathcal{S}$ to $\mathcal{S}$'s bottom level. Our selection procedure is shown in Algorithm 2. After prediction and learning at time t, our classifiers are given document set X_{t+1} at time $t+1$. Given a document $\mathbf{x}_{t+1} \in X_{t+1}$, we traverse the whole chunk structure $\mathcal{S}$ from root chunk $r_\mathcal{S}$ to leaves, and output the predicted classes that $\mathbf{x}_{t+1}$ belongs to. Parameters in discriminants $\mathcal{F}_{t+1,sc}$ are updated afterwards.

Algorithm 2: Greedy Selection via Chunk Structure $\mathcal{S}$

Input: $\mathcal{S}$, $\mathbf{x}_t$ $\mathbf{w}_{t-1} = \{\mathbf{w}_{t-1,sc}\}_{sc \in \mathcal{S}}$
$\mathbf{y} = \emptyset$;
for $sc = 1, 2, ..., SC$ **do**
  if $sc \in c_{y_{t,p(sc)}}$ **then**
    $\mathbf{y}_{sc} = \arg\max_{y \in \mathcal{Y}_{sc}, y \neq \mathbf{y}_{sc}} (w^T \Psi(\mathbf{x}_t, \mathbf{y}_{sc} \cup y))$;
  end
  if *sc is leaves chunk in* $\mathcal{S}$ **then**
    $\mathbf{y} = \mathbf{y} \cup \mathbf{y}_{sc}$;
  end
end
return *y*

5. EXPERIMENTAL SETUP

In §5.1, we propose 5 research questions to guide our experiments; we describe our dataset in §5.2 and set up our experiments in §5.3; §5.4 gives details about our evaluation metrics; the baselines are described in §5.5.

5.1 Research questions

We list the research questions, **RQ1** to **RQ5**, to guide the remainder of the paper.

RQ1 As a preliminary question, how does our chunk-based method perform in stationary HMC? (See §6.1)

RQ2 Is our document expansion strategy helpful for classifying documents in a HMC setting? (See §6.2)

RQ3 Does *concept drift* occur in our streaming short text collection? Does online topic extraction help to avoid *concept drift* on HMC-SST? (See §6.3)

RQ4 How does our proposed method perform on HMC-SST? Does it outperform baselines in terms of our evaluation metrics? (See §6.4)

RQ5 What is the effect of we change the size of chunks? Can we find an optimized value of the size of chunks in HMC-SST? (See §6.5)

5.2 Dataset

General statistics. We use a dataset of tweets related to a major public transportation system in a European country. The tweets were posted between January 18, 2010 and June 5, 2012, covering a period of nearly 30 months. The dataset includes 145, 692 tweets posted by 77,161 Twitter users. Using a state-of-the-art language identification tool [9], we found that over 95% tweets in our dataset is written in Dutch, whereas most other tweets are written in English. The dataset has human annotations for each tweet. A diverse set of social media experts produced the annotations after receiving proper training. In total, 81 annotators participated in the process.

The annotation tree for the dataset has 493 nodes. The annotations describe such aspects as reputation dimensions and product attributes and service. All annotators use Dutch during the annotating process. Unlike many other Twitter datasets with human annotations, e.g., Amigó et al. [2], in our dataset those labels are not independent from each other. Instead, each tweet is labeled by multiple hierarchical classes. From the root class, we divide the dataset into 13 individual subsets following the root node's child classes, which are shown in Table 1. In our experiment, not all subsets are included in our experiments: we ignore the subset with the fewest tweets: `Citizenship`. As all instances in `Online Source` are annotated by the same labels, we also omit it.

Author and temporal statistics. Fig. 5 shows the number of authors for different numbers of posted tweets in our dataset. Most

Table 1: The 13 subsets that make up our dataset, all annotations are in Dutch. The second column shows the English translation, the third column gives the number of tweets per subset, the fourth indicates whether a subset was included in our experiments.

Tag (in Dutch)	Translation	Number	Included
Berichtgeving	Communications	208,503	Yes
Aanbeveling	Recommendation	150,768	Yes
Bron online	Online source	2,505	No
Bron offline	Offline source	179,073	Yes
Reiziger	Type of traveler	123,281	Yes
Performance	Performance	28,545	Yes
Product	Product	82,284	Yes
Innovation	Innovation	114,647	Yes
Workplace	Workplace	16,910	Yes
Governance	Governance	11,340	Yes
Bedrijfsgerelateerd	Company related	15,715	Yes
Citizenship	Citizenship	628	No
Leadership	Leadership	10,410	Yes

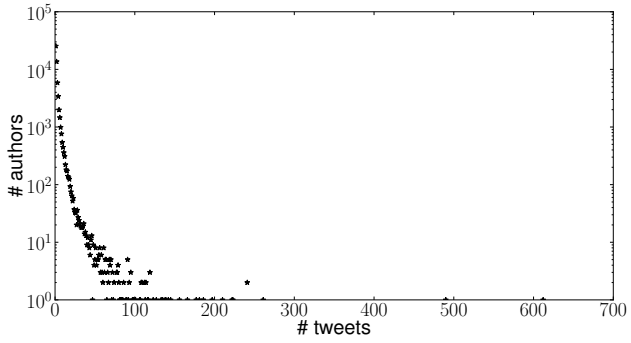


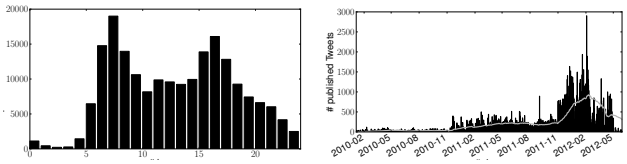
Figure 5: Number of tweets per user in our dataset, where the y-axis denotes the number of tweets and the x-axis denotes the corresponding number of tweets the author posted in our dataset. One user with more than 9000 tweets is omitted to improve readability.

users post fewer than 200 tweets. In our dataset, 73,245 users posts fewer than 10 tweets within the whole time period, and the maximum number of tweets posted by one user is 9,293: this is a news aggregator that accumulates and retweets information about public transportation systems.

One of the most interesting parts of the corpus is the possibility to analyze and test longitudinal temporal statistics. We can display the trends of tweets with various ways of binning. We can look at general developments over long periods of time and bin documents per day and per week. Fig. 6 shows the total number of tweets posted at each hour over 24 hours. Clearly, people commute in the train: the rush hours between 6am and 8am and between 4pm and 5pm correspond to a larger output of tweets. Fig. 6 also gives us statistics on the number of tweets posted per day; many more tweets are posted within the period from November 2011 to March 2012, and a peak of the number of tweets happening around February 18, 2012, a day with a lot of delays (according to the uttered tweets).

5.3 Experimental setup

Following [33], we set the hyper parameters $\alpha = 50/\left(K^g + K^l\right)$ and $\beta^l = \beta^g = 0.5$ in our experiments. We set $\lambda = 0.2$ and the number of samples to 5000 in our experiment for both document expansion and topic modeling. The number of topics in our topic modeling process is set to 50, for both Z_0^u and Z_0^{com}. For our chunk-based structural SVM classification, we set parameter $C = 0.0001$. For simplicity, we assume that each chunk in our experiments has at most 4 child nodes.



(a) Tweets per hour (b) Tweets per day

Figure 6: Number of tweets in our dataset. (Left): number of published tweets published per hour. (Right): number of published tweets published per day.

Statistical significance of observed differences between two comparisons is tested using a two-tailed paired t-test. In our experiments, statistical significance is denoted using ▲ (△) for strong (weak) significant differences for $\alpha = 0.01$ ($\alpha = 0.05$). For the stationary HMC evaluation, all experiments are executed using 10-fold cross validation combining training, validation and test sets.

5.4 Evaluation metrics

We adapt *precision* and *recall* to hierarchical multi-label learning following [4]. Given a class $i \in C$, let TP_i, FP_i and FN_i be the number of true positives, false positives and false negatives, respectively. Precision and recall for the whole output tree-structure are:

$$P = \frac{\sum_{i \in C} TP_i}{\sum_{i \in C} TP_i + \sum_{i \in C} FP_i}; \quad R = \frac{\sum_{i \in C} TP_i}{\sum_{i \in C} TP_i + \sum_{i \in C} FN_i} \tag{13}$$

We evaluate the performance using *macro* F_1*-measure* (combining precision and recall) and *average accuracy*. The *macro* F_1*-measure* measures the classification effectiveness for each individual class and averages them, whereas *average accuracy* measures the proportion correctly identified. For simplicity's sake, we abbreviate *average accuracy* as accuracy and acc. in §6.

5.5 Baselines and comparisons

We list the methods and baselines that we consider in Table 2. We write C-SSVM for the overall process as described in §4, which includes both document expansion and topic tracking. To be able to answer **RQ1**, we consider NDC-SSVM, which is C-SSVM without document expansion. Similarly, in the context of **RQ2** we consider GTC-SSVM and LTC-SSVM for variations of C-SSVM that only have global topics and local topics, respectively.

There are no previous methods that have been evaluated on the hierarchical multi-label classification of streaming short text. Because of this, we consider two types of baseline: stationary and streaming. For stationary hierarchical multi-label classification, we use CSSA, CLUS-HMC and H-SVM as baselines. We implement CSSA [4] by using kernel dependency estimation to reduce the possibly large number of labels to a manageable number of single-label learning problems. CLUS-HMC [42] is a method based on decision trees. H-SVM [14] extends normal SVMs to a hierarchical structure, where the SVM is trained in each node if, and only if, its parent node has been labeled positive. As CSSA and CLUS-HMC need to predefine the number of classes that each document belongs to, we employ MetaLabeler [40] to integrate with those two baselines.

Table 2: Baselines and methods used for comparison.

Acronym	Gloss	Reference
C-SSVM	Chunk-based structural learning method	This paper
NDC-SSVM	C-SSVM without document expansion	This paper
GTC-SSVM	C-SSVM only with global topics	This paper
LTC-SSVM	C-SSVM only with local topics	This paper
Stationary		
CSSA	Kernel density estimation based HMC method	[4]
CLUS-HMC	Decision tree-based HMC method	[42]
H-SVM	Hierarchical SVM for multi-label classification	[14]
Streaming		
H-SVM	Hierarchical SVM for multi-label classification	[14]
CSHC	Structural multi-class learning method	[12]
NBC	Naive Bayesian method	[21]

For the streaming short text classification task, besides H-SVM, we implement NBC and CSHC, a naive bayesian classifier framework, which has proved effective in streaming classification [21], and a structural multi-class learning method. Since NBC and CSHC are designed for single-label classification, we introduce a widely-used "one vs. all" strategy on multi-label situation [40]. We evaluate their performance after document expansion (§4.2)

6. RESULTS AND DISCUSSION

In §6.1, we compare C-SSVM to other baselines for stationary hierarchical multi-label classification; in §6.2 we examine the performance of document expansion. §6.3 details the effect of topic modeling on overcoming concept drift; §6.4 provides overall performance comparisons; §6.5 evaluates the influence of the number of items per chunk.

6.1 Performance on stationary HMC

We start by addressing **RQ1** and test if our C-SSVM is effective for the stationary HMC task, even though this is not the main purpose for which it was designed. Table 3 compares the macro F_1 of C-SSVM to the three HMC baselines. C-SSVM and CSSA tend to outperform the other baselines: for 6 out of 11 tags C-SSVM provides the best performance, while for the remaining 5 CSSA performs best. The performance differences between C-SSVM and CSSA are not statistically significant. This shows that, when compared against state of the art baselines in terms of the macro F_1 metric, C-SSVM is competitive.

6.2 Document expansion

Next, we turn to **RQ2** and evaluate the effectiveness of document expansion for HMC-SST. As described in §4, we extend a short text into a longer document by extracting sentences from linked Wikipedia articles. Table 4 shows an example of the document expansion where the new sentences are relevant to the original text.

Table 3: RQ1: macro F_1 values for stationary comparisons.

	C-SSVM	CSSA	CLUS-HMC	H-SVM
Communications	**0.5073**	0.5066	0.4812	0.4822
Recommendation	0.4543	**0.4612**	0.4421	0.4452
Offline source	**0.4245**	0.4176	0.4164	0.4161
Type of traveler	0.4623	**0.4677**	0.4652	0.4615
Performance	**0.5221**	0.5109	0.5054	0.5097
Product	**0.4762**	0.4722	0.4686	0.4609
Innovation	**0.4991**	0.4921	0.4822	0.4812
Workplace	0.4645	**0.4725**	0.4687	0.4623
Governance	0.4932	**0.5025**	0.4987	0.4923
Company related	0.4922	**0.4972**	0.4901	0.4852
Leadership	**0.4672**	0.4654	0.4624	0.4602

Table 4: An example of document expansion.

Short text
I'm tempted to get that LG Chocolate Touch. Or at least get a touchscreen phone

Extension
The original LG Chocolate KV5900 was released in Korea long before the UK or U.S. version.
The LG VX8500 or "Chocolate" is a slider cellphone-MP3 player hybrid that is sold as a feature phone.
The sensory information touch, pain, temperature etc., is then conveyed to the central nervous system by afferent neurones ...

Table 5: RQ2: Effect of document expansion in HMC.

	C-SSVM		NDC-SSVM	
Subset	macro-F_1	Acc.	macro-F_1	Acc.
Communication	**0.5073**$^{\blacktriangle}$	**0.5164**$^{\blacktriangle}$	0.4887	0.4972
Recommendation	**0.4543**	**0.4663**	0.4542	0.4655
Offline source	**0.4245**$^{\blacktriangle}$	**0.4523**$^{\blacktriangle}$	0.4112	0.4421
Type of traveler	0.4623	0.4731	**0.4647**	**0.4791**
Performance	**0.5221**$^{\blacktriangle}$	**0.5321**$^{\blacktriangle}$	0.5013	0.5111
Product	**0.4762**$^{\triangle}$	**0.4823**$^{\triangle}$	0.4612	0.4721
Innovation	**0.4991**$^{\blacktriangle}$	**0.5121**$^{\blacktriangle}$	0.4522	0.4612
Workplace	**0.4645**$^{\triangle}$	**0.4724**$^{\triangle}$	0.4601	0.4695
Governance	**0.4932**$^{\blacktriangle}$	**0.5072**$^{\blacktriangle}$	0.4787	0.4944
Company related	**0.4922**$^{\blacktriangle}$	**0.5072**$^{\blacktriangle}$	0.4772	0.4921
Leadership	**0.4672**$^{\triangle}$	**0.4754**	0.4601	0.4707

Table 5 contrasts the evaluation results for C-SSVM with that of NDC-SSVM, which excludes documents expansion, in terms of macro-F_1 and average accuracy. We find that C-SSVM outperforms NDC-SSVM for most subsets of stationary HMC comparisons. In terms of macro F_1, C-SSVM offers an increase over NDC-SSVM of up to 9.4%, whereas average accuracy increases by up to 9.9% significantly. We conclude that document expansion is effective for the stationary HMC task, especially for short text classification.

6.3 Time-aware topic extraction

Our third research question **RQ3** aims at determining whether concept drift occurs and whether topic extraction helps to avoid this. Fig. 7 shows the propagation process of an example local topic for the subset "Communication." The upper part of Fig. 7 shows the 5 most representative terms for the topic during 5 time periods. The bottom half of the figure plots fluctuating topical distributions over time, which indicates concept drift between two adjacent periods.

Fig. 8 shows the macro F_1 score over time for C-SSVM, C-SSVM with only local topics (LTC-SSVM), and C-SSVM with only globale topics (GTC-SSVM). This helps us understand whether C-SSVM is able to deal with concept drift during classification. We see that the performance in terms of macro F_1 increases over time, rapidly in the early stages, more slowly in the later periods covered by our data set, while not actually plateauing. We also see that the performance curves of LTC-SSVM and GTC-SSVM behave similarly, albeit at a lower performance level. Between LTC-SSVM and GTC-SSVM, LTC-SSVM outperforms GTC-SSVM slightly: local

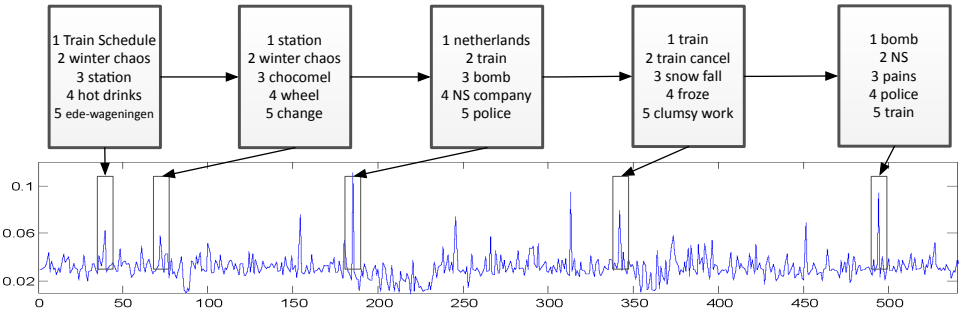


Figure 7: RQ3: An example local topic propagation in the subset "Communication." The text blocks at the top indicate the top 5 representative terms for the topic being propagated at a specific time period; the bottom side shows the topic distribution over the whole timeline.

topic distributions are more sensitive, and hence adaptive, when drift occurs.

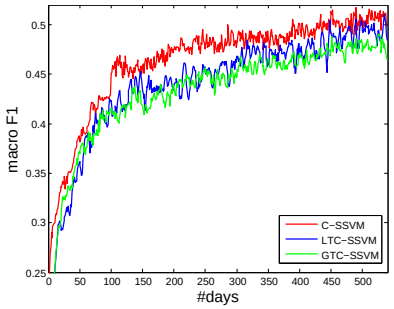
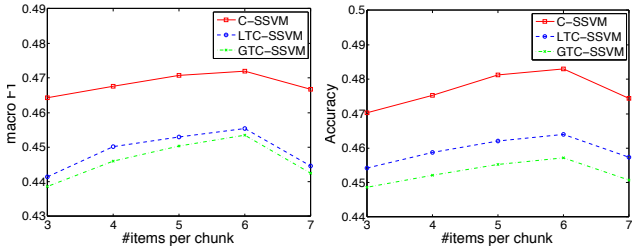


Figure 8: RQ3: macro F_1 performance of C-SSVM, LTC-SSVM and GTC-SSVM over the entire data set.

6.4 Overall comparison

To help us answer **RQ4**, Table 6 lists the macro F_1 and average accuracy for all methods listed in Table 2 for all subsets over all time periods. We see that our proposed methods C-SSVM, NDC-SSVM, GTC-SSVM and LTC-SSVM significantly outperform the baselines on most of subsets.

As predicted, NBC performs worse. Using local topics (LTC-SSVM) performs second best (after using both local and global topics), which indicates the importance of dynamic local topics tracking in our streaming classification. C-SSVM achieves a 3.2% (4.5%) increase over GTC-SSVM in terms of macro F_1 (accuracy), whereas the macro F_1 (accuracy) increases 1.9% (2.2%) over LTC-SSVM. Compared to CSHC, C-SSVM offers a statistically significant improvement of up to 7.6% and 8.1% in terms of macro F_1 and accuracy, respectively.



(a) macro F_1 (b) Accuracy

Figure 9: RQ5: Performance with different numbers of items of each chunk, in terms of macro F_1 (a) and Accuracy (b).

6.5 Chunks

We now move on to **RQ5**, and analyse the influence of the number of items per chunk. Fig. 9 plots the performance curves for C-SSVM, LTC-SSVM and GTC-SSVM with varying numbers of items per chunk. While not statistically significant, for both metrics and all three methods, the performance peaks when the number of items equals 6, i.e., higher than our default value of 4.

7. CONCLUSION AND FUTURE WORK

We considered the task of hierarchical multi-label classification of social text streams. We identified three main challenges: the shortness of text, concept drift, and hierarchical labels as classification targets. The first of these was tackled using an entity-based document expansion strategy. To alleviate the phenomenon of concept drift we presented a dynamic extension to topic models. This extension tracks topics with concept drift over time, based on both local and global topic distributions. We combine this with an innovative chunk-based structural learning framework to tackle the hierarchical multi-label classification problem. We verified the effectiveness of our proposed method in hierarchical multi-label classification of social text streams, showing significant improvements over various baselines tested with a manually annotated dataset of tweets.

As to future work, parallel processing may enhance the efficiency of our method on hierarchical multi-label classification of social text streams. Meanwhile, both the transfer of our approach to a larger social documents dataset and new baselines for document expansion and topic modeling should give new insights. Adaptive learning or semi-supervised learning can be used to optimize the chunk size in our task. Finally, we have evaluated our approaches on fixed time intervals. This might not accurately reflect exact concept drift on social streams. A novel incremental classification method focussing on dynamic time bins opens another direction of future research.

Acknowledgments. This research was supported by the China Scholarship Council, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nrs 288024 and 312827, the Netherlands Organisation for Scientific Research (NWO) under project nrs 727.-011.005, 612.001.116, HOR-11-10, 640.006.013, the Center for Creation, Content and Technology (CCCT), the QuaMerdes project funded by the CLARIN-nl program, the TROVe project funded by the CLARIAH program, the Dutch national program COMMIT, the ESF Research Network Program ELIAS, the Elite Network Shifts project funded by the Royal Dutch Academy of Sciences (KNAW), the Netherlands eScience Center under project number 027.012.105, the Yahoo! Faculty Research and Engagement Program, the Microsoft Research PhD program, and the HPC Fund.

Table 6: RQ4: Performance of all methods on all subsets for all time periods; macro F_1 is abbreviated to m-F_1, average accuracy is written as Acc. We use $^{\blacktriangle}$ and $^{\triangle}$ to denote significant improvements over CSHC. Best performance per subset is indicated in boldface.

	C-SSVM		NDC-SSVM		GTC-SSVM		LTC-SSVM		CSHC		H-SVM		NBC	
Subset	m-F_1	Acc.	m-F_1	Acc.	m-F_1	Acc.	m-F_1	Acc.	m-F_1	Acc.	m-F_1	Acc.	m-F_1	Acc.
Communication	**47.21**$^{\blacktriangle}$	**48.16**$^{\blacktriangle}$	44.24	45.42	46.44$^{\blacktriangle}$	47.68$^{\blacktriangle}$	46.25$^{\blacktriangle}$	47.82$^{\blacktriangle}$	44.12	45.31	45.22	46.62	44.02	45.18
Recommendation	**41.28**$^{\blacktriangle}$	**42.52**$^{\blacktriangle}$	40.44$^{\blacktriangle}$	41.52$^{\blacktriangle}$	39.88$^{\triangle}$	40.24$^{\triangle}$	40.52$^{\blacktriangle}$	41.47$^{\blacktriangle}$	38.53	39.42	38.22	39.71	34.31	35.26
Offline source	**40.69**$^{\blacktriangle}$	**41.61**$^{\blacktriangle}$	39.52$^{\blacktriangle}$	40.42$^{\blacktriangle}$	39.62$^{\blacktriangle}$	41.15$^{\blacktriangle}$	40.33$^{\blacktriangle}$	41.72$^{\blacktriangle}$	36.98	37.43	37.41	38.42	33.21	34.51
Type of traveler	43.73$^{\blacktriangle}$	44.61$^{\blacktriangle}$	**44.02**$^{\blacktriangle}$	**44.96**$^{\blacktriangle}$	43.12$^{\blacktriangle}$	44.25$^{\blacktriangle}$	43.45$^{\blacktriangle}$	44.49$^{\blacktriangle}$	38.83	40.01	41.07	41.92	38.62	39.38
Performance	**49.52**$^{\triangle}$	**50.81**$^{\triangle}$	47.62	48.45	48.86	49.63	48.93	50.02	48.74	49.26	48.84	49.52	46.42	47.32
Product	**44.88**$^{\blacktriangle}$	**45.24**$^{\blacktriangle}$	43.16$^{\blacktriangle}$	44.09$^{\blacktriangle}$	44.26$^{\blacktriangle}$	45.02$^{\blacktriangle}$	44.01$^{\blacktriangle}$	45.22$^{\blacktriangle}$	41.92	42.85	41.55	42.34	39.21	40.42
Innovation	**46.89**$^{\triangle}$	**47.68**$^{\triangle}$	45.58	46.64	45.97	46.81	46.52$^{\triangle}$	47.51$^{\triangle}$	45.44	46.56	44.52	45.63	43.41	44.21
Workplace	**43.81**$^{\blacktriangle}$	**44.42**$^{\blacktriangle}$	43.11$^{\blacktriangle}$	44.32$^{\blacktriangle}$	42.21$^{\blacktriangle}$	43.15$^{\blacktriangle}$	42.63$^{\blacktriangle}$	43.41$^{\blacktriangle}$	36.94	37.22	36.24	37.01	36.59	37.41
Governance	**47.71**$^{\blacktriangle}$	**48.44**$^{\blacktriangle}$	47.19$^{\blacktriangle}$	48.46$^{\blacktriangle}$	46.42$^{\triangle}$	47.35$^{\triangle}$	47.22$^{\triangle}$	48.19$^{\triangle}$	45.61	46.21	46.25	47.36	43.48	44.51
Company related	**47.20**$^{\blacktriangle}$	**48.52**$^{\blacktriangle}$	46.52$^{\blacktriangle}$	47.38$^{\blacktriangle}$	46.12$^{\blacktriangle}$	47.51$^{\blacktriangle}$	46.54$^{\blacktriangle}$	47.43$^{\blacktriangle}$	43.31	44.99	43.06	44.12	40.91	41.75
Leadership	**44.15**$^{\triangle}$	**45.88**$^{\blacktriangle}$	43.67	44.59	41.75	42.82	42.34	43.21	42.51	43.44	42.15	43.51	40.35	41.27

8. REFERENCES

[1] B. Albert, G. Joao, P. Mykola, and Z. Indre. Handling concept drift: importance challenges and solutions. In *PAKDD*, 2011.

[2] E. Amigó, A. Corujo, J. Gonzalo, E. Meij, and M. de Rijke. Overview of RepLab 2012: Evaluating online reputation management systems. In *CLEF*, 2012.

[3] Z. Barutcuoglu, R. Schapire, and O. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 2006.

[4] W. Bi and J. T. Kwok. Multi-label classification on tree-and dag-structured hierarchies. In *ICML*, 2011.

[5] D. Blei and J. Lafferty. Dynamic topic models. In *ICML*, 2006.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 2003.

[7] H. Blockeel, L. Schietgat, and J. Struyf. Decision trees for hierarchical multilabel classification. In *ECML*, 2006.

[8] D. Bollegala, Y. Matsuo, and M. Ishizuka. Measuring semantic similarity between words using web search engines. *WWW*, 2007.

[9] S. Carter, W. Weerkamp, and M. Tsagkias. Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text. *LREC*, 2013.

[10] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Incremental algorithms for hierarchical classification. *JMLR*, 2006.

[11] W. Chan, W. Yang, J. Tang, J. Du, X. Zhou, and W. Wang. Community question topic categorization via hierarchical kernelized classification. In *CIKM*, 2013.

[12] J. Chen and D. Warren. Cost-sensitive learning for large-scale hierarchical classification of commercial products. In *CIKM*, 2013.

[13] M. Chen, X. Jin, and D. Shen. Short text classification improved by learning multi-granularity topics. In *IJCAI*, 2011.

[14] A. Clare. *Machine Learning and Data Mining for Yeast Functional Genomics*. PhD thesis, University of Wales, 2003.

[15] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell.*, 2004.

[16] T. Finley and T. Joachims. Training structural svms when exact inference is intractable. In *ICML*, 2008.

[17] G. P. C. Fung, J. X. Yu, and H. Lu. Classifying text streams in the presence of concept drifting. In *PAKDD*, 2004.

[18] Y. Guo and S. Gu. Multi-label classification using conditional dependency networks. In *IJCAI*, 2011.

[19] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In *ICML*, 1997.

[20] X. Kong, B. Cao, and P. S. Yu. Multi-label classification by mining label and instance correlations from heterogeneous information networks. In *KDD*, 2013.

[21] G. Lebanon and Y. Zhao. Local likelihood modeling of temporal text streams. In *ICML*, 2008.

[22] L. Li, K. Zhou, G.-R. Xue, H. Zha, and Y. Yu. Enhancing diversity, coverage and balance for summarization through structure learning. In *WWW*, 2009.

[23] L. Li, K. Zhou, G.-R. Xue, H. Zha, and Y. Yu. Video summarization via transferrable structured learning. In *WWW*, 2011.

[24] J. S. Liu. The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *JASA*, 1994.

[25] S. Liu, S. Wang, F. Zhu, J. Zhang, and R. Krishnan. Hydra: Large-scale social identity linkage via heterogeneous behavior modeling. In *SIGMOD*, 2014.

[26] G. Long, L. Chen, X. Zhu, and C. Zhang. Tcsst: transfer classification of short & sparse text using external data. In *CIKM*, 2012.

[27] E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. In *WSDM 2012*, 2012.

[28] K. Nishida, R. Banno, K. Fujimura, and T. Hoshide. Tweet classification by data compression. In *DETECT*, 2011.

[29] K. Nishida, T. Hoshide, and K. Fujimura. Improving tweet stream classification by detecting changes in word probability. In *SIGIR*, 2012.

[30] D. Odijk, E. Meij, and M. de Rijke. Feeding the second screen: Semantic linking based on subtitles. In *OAIR*, 2013.

[31] J. Petterson and T. S. Caetano. Submodular multi-label learning. In *NIPS*, 2011.

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

[33] Z. Ren, S. Liang, E. Meij, and M. de Rijke. Personalized time-aware tweets summarization. In *SIGIR*, 2013.

[34] J. Rousu, S. Saunders, S. Szedmak, and J. Shawe-Taylor. Kernel-based learning of hierarchical multilabel classification models. *Mach. Learn.*, 2006.

[35] S. Sarawagi and R. Gupta. Accurate max-margin training for structured output spaces. In *ICML*, 2008.

[36] S. Shalev-Shwartz and Y. Singer. Efficient learning of label ranking by soft projections onto polyhedra. *JMLR*, 2006.

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

[38] A. Sun. Short text classification using very few words. In *SIGIR*, 2012.

[39] N. A. Syed, H. Liu, and K. K. Sung. Handling concept drifts in incremental learning with support vector machines. In *KDD*, 1999.

[40] L. Tang, S. Rajan, and V. K. Narayanan. Large-scale mutli-label classification via metalabeler. In *WWW*, 2009.

[41] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 2005.

[42] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel. Decision trees for hierarchical multi-label classification. *JMLR*, 2008.

[43] X. Wan and J. Yang. Multi-document summarization using cluster-based link analysis. In *SIGIR*, 2008.

[44] W.-T. Yih and C. Meek. Improving similarity measures for short segments of text. In *AAAI*, 2007.

[45] Y. Yue and T. Joachims. Predicting diverse subsets using structural SVMs. In *ICML*, 2008.

[46] S. Zhang, X. Jin, D. Shen, B. Cao, X. Ding, and X. Zhang. Short text classification by detecting information path. In *CIKM*, 2013.