Tweet Segmentation and its Application to Named Entity Recognition

Chenliang Li, Aixin Sun, Jianshu Weng, and Qi He

Abstract—Twitter has attracted millions of users to share and disseminate most up-to-date information, resulting in large volumes of data produced everyday. However, many applications in Information Retrieval (IR) and Natural Language Processing (NLP) suffer severely from the noisy and short nature of tweets. In this paper, we propose a novel framework for tweet segmentation in a batch mode, called *HybridSeg*. By splitting tweets into meaningful segments, the semantic or context information is well preserved and easily extracted by the downstream applications. *HybridSeg* finds the optimal segmentation of a tweet by maximizing the sum of the stickiness scores of its candidate segments. The stickiness score considers the probability of a segment being a phrase in English (*i.e.,* global context) and the probability of a segment being a phrase within the batch of tweets (*i.e.,* local context). For the latter, we propose and evaluate two models to derive local context by considering the linguistic features and term-dependency in a batch of tweets, respectively. *HybridSeg* is also designed to iteratively learn from confident segments as pseudo feedback. Experiments on two tweet datasets show that tweet segmentation quality is significantly improved by learning both global and local contexts compared with using global context alone. Through analysis and comparison, we show that local linguistic features are more reliable for learning local context compared with term-dependency. As an application, we show that high accuracy is achieved in named entity recognition by applying segment-based part-of-speech (POS) tagging.

Index Terms—Twitter Stream, Tweet Segmentation, Named Entity Recognition, Linguistic Processing, Wikipedia

1 Introduction

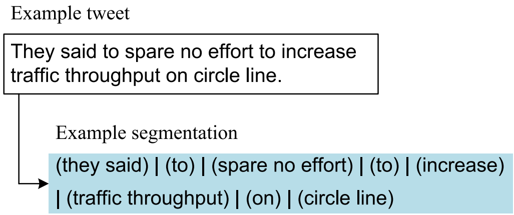
MICROBLOGGING sites such as Twitter have reshaped the way people find, share, and disseminate timely information. Many organizations have been reported to create and monitor targeted Twitter streams to collect and understand users' opinions. Targeted Twitter stream is usually constructed by filtering tweets with predefined selection criteria (*e.g.,* tweets published by users from a geographical region, tweets that match one or more predefined keywords). Due to its invaluable business value of timely information from these tweets, it is imperative to understand tweets' language for a large body of downstream applications, such as named entity recognition (NER) [1], [3], [4], event detection and summarization [5], [6], [7], opinion mining [8], [9], sentiment analysis [10], [11], and many others.

Given the limited length of a tweet (*i.e.,* 140 characters) and no restrictions on its writing styles, tweets often contain grammatical errors, misspellings, and informal abbreviations. The error-prone and short nature of tweets often make the word-level language models for tweets less reliable. For example, given a tweet "I call her, no answer. Her phone in the bag, she dancin.", there is no clue to guess its true theme by disregarding word order (*i.e.,* bag-of-word model). The situation is further exacerbated with the limited context provided by the tweet. That is, more than one explanation for this tweet could be derived by different readers if the tweet is considered in isolation. On the other hand, despite the noisy nature of tweets, the core semantic information is well preserved in tweets in the form of named entities or semantic phrases. For example, the emerging phrase "she dancin" in the related tweets indicates that it is a key concept – it classifies this tweet into the family of tweets talking about the song "She Dancin", a trend topic in Bay Area in Jan, 2013.

In this paper, we focus on the task of *tweet segmentation*. The goal of this task is to split a tweet into a sequence of *consecutive n-grams* ($n \geq 1$), each of which is called a *segment*. A segment can be a named entity (*e.g.,* a movie title "finding nemo"), a semantically meaningful information unit (*e.g.,* "officially released"), or any other types of phrases which appear "more than by chance" [1]. Figure 1 gives an example. In this example, a tweet "*They said to spare no effort to increase traffic throughput on circle line.*" is split into eight segments. Semantically meaningful segments "spare on effort", "traffic throughput" and "circle line" are preserved. Because these segments preserve semantic meaning of the tweet more precisely than each of its constituent words does, the topic of this tweet can be better captured in the subsequent processing of this tweet. For instance, this segment-based representation could be used to enhance the extrac-

This paper is an extended version of two SIGIR conference papers [1], [2].

- *C. Li is with State Key Lab of Software Engineering, School of Computer, Wuhan University, China. E-Mail: cllee@whu.edu.cn*
- *A. Sun is with School of Computer Engineering, Nanyang Technological University, Singapore. E-Mail: axsun@ntu.edu.sg*
- *J. Weng is an independent researcher, Singapore. E-Mail: jianshu@acm.org*
- *Q. He is with LinkedIn Inc. E-Mail: qhe@linkedin.com*



Example tweet

They said to spare no effort to increase traffic throughput on circle line.

Example segmentation

(they said) | (to) | (spare no effort) | (to) | (increase) | (traffic throughput) | (on) | (circle line)

Fig. 1: Example of Tweet Segmentation

tion of geographical location from tweets because of the segment "circle line" [12]. In fact, segment-based representation has shown its effectiveness over word-based representation in the tasks of named entity recognition and event detection [1], [2], [13]. Note that, a named entity is valid segment; but a segment may not necessarily be a named entity. In [6] the segment "korea vs greece" is detected for the event related to the world cup match between Korea and Greece.

To achieve high quality tweet segmentation, we propose a generic tweet segmentation framework, named *HybridSeg*. *HybridSeg* learns from both *global* and *local* contexts, and has the ability of learning from *pseudo feedback*.

Global context. Tweets are posted for information sharing and communication. The named entities and semantic phrases are well preserved in tweets. The global context derived from Web pages (*e.g.*, Microsoft Web N-Gram corpus) or Wikipedia therefore helps identifying the meaningful segments in tweets. The method realizing the proposed framework that solely relies on global context is denoted by $\mathit{HybridSeg}_{Web}$.

Local context. Tweets are highly time-sensitive so that many emerging phrases like "She Dancin" cannot be found in external knowledge bases. However, considering a large number of tweets published within a short time period (*e.g.*, a day) containing the phrase, it is not difficult to recognize "She Dancin" as a valid and meaningful segment. We therefore investigate two local contexts, namely local linguistic features and local collocation. Observe that tweets from many official accounts of news agencies, organizations, and advertisers are likely well written. The well preserved linguistic features in these tweets facilitate named entity recognition with high accuracy. Each named entity is a valid segment. The method utilizing local linguistic features is denoted by $\mathit{HybridSeg}_{NER}$. It obtains confident segments based on the voting results of multiple off-the-shelf NER tools. Another method utilizing local collocation knowledge, denoted by $\mathit{HybridSeg}_{NGram}$, is proposed based on the observation that many tweets published within a short time period are about the same topic. $\mathit{HybridSeg}_{NGram}$ segments tweets by estimating the term-dependency within a batch of tweets.

Pseudo feedback. The segments recognized based on local context with high confidence serve as good feedback to extract more meaningful segments. The learning from pseudo feedback is conducted iteratively and the method implementing the iterative learning is named $\mathit{HybridSeg}_{Iter}$.

We conduct extensive experimental analysis on *HybridSeg*[1] on two tweet datasets and evaluate the quality of tweet segmentation against manually annotated tweets. Our experimental results show that $\mathit{HybridSeg}_{NER}$ and $\mathit{HybridSeg}_{NGram}$, the two methods incorporating local context in additional to global context, achieve significant improvement in segmentation quality over $\mathit{HybridSeg}_{Web}$, the method use global context alone. Between the former two methods, $\mathit{HybridSeg}_{NER}$ is less sensitive to parameter settings than $\mathit{HybridSeg}_{NGram}$ and achieves better segmentation quality. With iterative learning from pseudo feedback, $\mathit{HybridSeg}_{Iter}$ further improves the segmentation quality.

As an application of tweet segmentation, we propose and evaluate two segment-based NER algorithms. Both algorithms are unsupervised in nature and take tweet segments as input. One algorithm exploits co-occurrence of named entities in targeted Twitter streams by applying random walk (RW) with the assumption that named entities are more likely to co-occur together. The other algorithm utilizes Part-of-Speech (POS) tags of the constituent words in segments. The segments that are likely to be a noun phrase are considered as named entities. Our experimental results show that (i) the quality of tweet segmentation significantly affects the accuracy of N-ER, and (ii) POS-based NER method outperforms RW-based method on both datasets.

The rest of this paper is organized as follows. Section 2 surveys related works on tweet segmentation. Section 3 defines tweet segmentation and describes the proposed framework. Section 4 details how the local context is exploited in the framework. In Section 5, the segment-based NER methods are investigated. In Section 6, we evaluate the proposed *HybridSeg* framework and the two segment-based NER methods. Section 7 concludes this paper.

2 Related Work

Both tweet segmentation and named entity recognition are considered important subtasks in NLP. Many existing NLP techniques heavily rely on linguistic features, such as POS tags of the surrounding words, word capitalization, trigger words (*e.g.*, Mr., Dr.), and gazetteers. These linguistic features, together with effective supervised learning algorithms (*e.g.*, hidden markov model (HMM) and conditional random field (CRF)), achieve very good performance on formal text

[1] *HybridSeg* refers to $\mathit{HybridSeg}_{Web}$, $\mathit{HybridSeg}_{NER}$, $\mathit{HybridSeg}_{NGram}$ and $\mathit{HybridSeg}_{Iter}$ or one of them based on the context. We do not distinguish this when the context is clear and discriminative.

corpus [14], [15], [16]. However, these techniques experience severe performance deterioration on tweets because of the noisy and short nature of the latter.

There have been a lot of attempts to incorporate tweet's unique characteristics into the conventional NLP techniques. To improve POS tagging on tweets, Ritter *et al.* train a POS tagger by using CRF model with conventional and tweet-specific features [3]. Brown clustering is applied in their work to deal with the ill-formed words. Gimple *et al.* incorporate tweet-specific features including at-mentions, hashtags, URLs, and emotions [17] with the help of a new labeling scheme. In their approach, they measure the confidence of capitalized words and apply phonetic normalization to ill-formed words to address possible peculiar writings in tweets. It was reported to outperform the state-of-the-art Stanford POS tagger on tweets. Normalization of ill-formed words in tweets has established itself as an important research problem [18]. A supervised approach is employed in [18] to first identify the ill-formed words. Then, the correct normalization of the ill-formed word is selected based on a number of lexical similarity measures.

Both supervised and unsupervised approaches have been proposed for named entity recognition in tweets. T-NER, a part of the tweet-specific NLP framework in [3], first segments named entities using a CRF model with orthographic, contextual, dictionary and tweet-specific features. It then labels the named entities by applying Labeled-LDA with the external knowledge base Freebase.[2] The NER solution proposed in [4] is also based on a CRF model. It is a two-stage prediction aggregation model. In the first stage, a KNN-based classifier is used to conduct word-level classification, leveraging the similar and recently labeled tweets. In the second stage, those predictions, along with other linguistic features, are fed into a CRF model for finer-grained classification. Chua *et al.* [19] propose to extract noun phrases from tweets using an unsupervised approach which is mainly based on POS tagging. Each extracted noun phrase is a candidate named entity.

Our work is also related to entity linking (EL). EL is to identify the mention of a named entity and link it to an entry in a knowledge base like Wikipedia [20], [21], [22], [23]. Conventionally, EL involves a NER system followed by a linking system [20], [21]. Recently, Sil and Yates propose to combine named entity recognition and linking into a joint model [23]. Similarly, Guo *et al.* propose a structural SVM solution to simultaneously recognize mention and resolve the linking [22]. While entity linking aims to identify the boundary of a named entity and resolve its meaning based on an external knowledge base, a typical NER system identifies entity mentions only, like the work presented here. It is difficult to make a fair comparison between these two techniques.

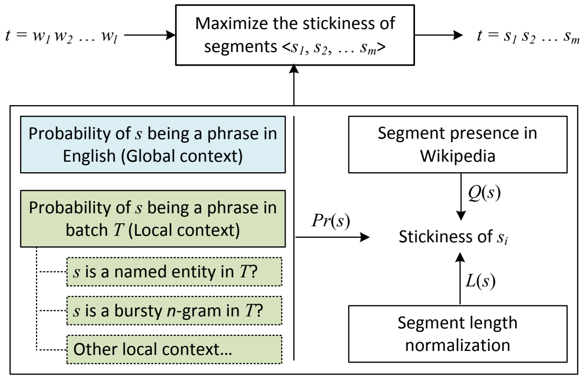

t = w1 w2 … wl
Maximize the stickiness of segments <s1, s2, … sm>
t = s1 s2 … sm
Probability of s being a phrase in English (Global context)
Probability of s being a phrase in batch T (Local context)
s is a named entity in T?
s is a bursty n-gram in T?
Other local context…
Pr(s)
Segment presence in Wikipedia
Q(s)
Stickiness of si
L(s)
Segment length normalization

Fig. 2: *HybridSeg* framework without learning from pseudo feedback

Tweet segmentation is conceptually similar to Chinese word segmentation (CSW). Text in Chinese is a continuous sequence of characters. Segmenting the sequence into meaningful words is the first step in most applications. State-of-the-art CSW methods are mostly developed using supervised learning techniques like perceptron learning and CRF model [24], [25], [26], [27], [28]. Both linguistic and lexicon features are used in the supervised learning in CSW. Tweets are extremely noisy with misspellings, informal abbreviations, and grammatical errors. These adverse properties lead to a huge number of training samples for applying a supervised learning technique. Here, we exploit the semantic information of external knowledge bases and local contexts to recognize meaningful segments like named entities and semantic phrases in Tweets. Very recently, similar idea has also been explored for CSW by Jiang *et al.* [28]. They propose to prune the search space in CSW by exploiting the natural annotations in the Web. Their experimental results show significant improvement by using simple local features.

3 *HybridSeg* FRAMEWORK

The proposed *HybridSeg* framework segments tweets in batch mode. Tweets from a targeted Twitter stream are grouped into batches by their publication time using a fixed time interval (*e.g.*, a day). Each batch of tweets are then segmented by *HybridSeg* collectively.

3.1 Tweet Segmentation

Given a tweet t from batch $\mathcal{T}$, the problem of tweet segmentation is to split the ℓ words in $t = w_1 w_2 \dots w_\ell$ into $m \leq \ell$ consecutive segments, $t = s_1 s_2 ... s_m$, where each segment s_i contains one or more words. We formulate the tweet segmentation problem as an optimization problem to maximize the sum of *stickiness* scores of the m segments, shown in Figure 2.[3] A high

[2] http://www.freebase.com/

[3] For clarity, we do not show the iterative learning from pseudo feedback in this figure.

stickiness score of segment s indicates that it is a phrase which appears "more than by chance", and further splitting it could break the correct word collocation or the semantic meaning of the phrase. Formally, let $\mathcal{C}(s)$ denote the *stickiness* function of segment s. The optimal segmentation is defined in the following:

$$\arg\max_{s_1,\dots,s_m} \sum_{i=1}^{m} \mathcal{C}(s_i) \tag{1}$$

The optimal segmentation can be derived by using dynamic programming with a time complexity of $O(\ell)$ (rf. [1] for detail).

As shown in Figure 2, the *stickiness* function of a segment takes in three factors: (i) length normalization $L(s)$, (ii) the segment's presence in Wikipedia $Q(s)$, and (iii) the segment's phraseness $Pr(s)$, or the probability of s being a phrase based on global and local contexts. The stickiness of s, $\mathcal{C}(s)$, is formally defined in Eq. 2, which captures the three factors:

$$\mathcal{C}(s) = \mathcal{L}(s) \cdot e^{Q(s)} \cdot \frac{2}{1+e^{-SCP(s)}} \tag{2}$$

Length normalization. As the key of tweet segmentation is to extract meaningful phrases, longer segments are preferred for preserving more topically specific meanings. Let $|s|$ be number of words in segment s. The normalized segment length $\mathcal{L}(s) = 1$ if $|s| = 1$ and $\mathcal{L}(s) = \frac{|s|-1}{|s|}$ if $|s| > 1$, which moderately alleviates the penalty on long segments.

Presence in Wikipedia. In our framework, Wikipedia serves as an external dictionary of valid names or phrases. Specifically, $Q(s)$ in Eq. 2 is the probability that s is an anchor text in Wikipedia, also known as *keyphraseness* in [21], [29]. Let $wiki(s)$ and $wiki_a(s)$ be the number of Wikipedia entries where s appears in any form and s appears in the form of anchor text, respectively, $Q(s) = wiki_a(s)/wiki(s)$. Each anchor text in Wikipedia refers to a Wikipedia entry even if the entry has not been created. The segment that is often used as anchor text in Wikipedia is preferred in our segmentation. Note that Wikipedia here can be replaced with any other external knowledge base by redefining $Q(s)$. Example knowledge bases include Freebase, Probase [30], or domain-specific knowledge base like GeoNames[4] if the targeted Twitter stream is domain-specific.

Segment phraseness. The last component of Eq. 2 is to estimate the probability of a segment being a valid phrase using *Symmetric Conditional Probability* (SCP) measure,[5] defined in Eq. 3.

$$SCP(s) = \log \frac{\Pr(s)^2}{\frac{1}{|s|-1}\sum_{i=1}^{|s|-1} \Pr(w_1 \dots w_i)\Pr(w_{i+1} \dots w_{|s|})} \tag{3}$$

In Eq. 3, $Pr(s)$ or $Pr(w_1 \dots w_i)$ is the approximated n-gram probability of a segment. If s contains a single word w, $SCP(s) = 2\log Pr(w)$.

The estimation of $Pr(s)$ is the key challenge in our framework. In the following, we present three observations, which are also the rationales why $Pr(s)$ can be estimated from global and local contexts.

3.2 Observations for Tweet Segmentation

Tweets are considered noisy with lots of informal abbreviations and grammatical errors. However, tweets are posted mainly for information sharing and communication among many purposes.

Observation 1: Word collocations of named entities and common phrases in English are well preserved in Tweets.
Many named entities and common phrases are preserved in tweets for information sharing and dissemination. In this sense, $Pr(s)$ can be estimated by counting a segment's appearances in a very large English corpus (*i.e.,* global context). In our implementation, we turn to Microsoft Web N-Gram corpus [31]. This N-Gram corpus is derived from all Web documents indexed by Microsoft Bing in the EN-US market. It provides a good estimate of the statistics of commonly used phrases in English.

Observation 2: Many tweets contain useful linguistic features.
Although many tweets contain unreliable linguistic features like misspellings and unreliable capitalizations [3], there exist tweets composed in proper English. For example, tweets published by official accounts of news agencies, organizations, and advertisers are often well written. The linguistic features in these tweets enable named entity recognition with relatively high accuracy.

Observation 3: Tweets in a targeted stream are not topically independent to each other within a time window.
Many tweets published within a short time period talk about the same theme. These similar tweets largely share the same segments. For example, similar tweets have been grouped together to collectively detect events, and an event can be represented by the common discriminative segments across tweets [13].

The latter two observations essentially reveal the same phenomenon: local context in a batch of tweets complements global context in segmenting tweets. For

[4] http://www.geonames.org/

[5] In our earlier work [1], we have evaluated two collocation measures, SCP and *Pointwise Mutual Information* (PMI). Our experimental results show that SCP is much more effective than PMI for tweet segmentation.

example, person names emerging from bursty events may not be recorded in Wikipedia. However, if the names are reported in tweets by news agencies or mentioned in many tweets, there is a good chance to segment these names correctly based on local linguistic features or local word collocation from the batch of tweets. In the next section, we detail learning from local context to estimate $Pr(s)$.

4 Learning from Local Context

Illustrated in Figure 2, the segment phraseness $Pr(s)$ is computed based on both global and local contexts. Based on Observation 1, $Pr(s)$ is estimated using the n-gram probability provided by Microsoft Web N-Gram service, derived from English Web pages. We now detail the estimation of $Pr(s)$ by learning from local context based on Observations 2 and 3. Specifically, we propose learning $Pr(s)$ from the results of using off-the-shelf Named Entity Recognizers (NERs), and learning $Pr(s)$ from local word collocation in a batch of tweets. The two corresponding methods utilizing the local context are denoted by $HybridSeg_{NER}$ and $HybridSeg_{NGram}$ respectively.

4.1 Learning from Weak NERs

To leverage the local linguistic features of well-written tweets, we apply multiple off-the-shelf NERs trained on formal texts to detect named entities in a batch of tweets $\mathcal{T}$ by voting. Voting by multiple NERs partially alleviates the errors due to noise in tweets. Because these NERs are not specifically trained on tweets, we also call them weak NERs. Recall that each named entity is a valid segment, the detected named entities are valid segments.

Given a candidate segment s, let f_s be its total frequency in $\mathcal{T}$. A NER r_i may recognize s as a named entity $f_{r_i,s}$ times. Note that $f_{r_i,s} \leq f_s$ since a NER may only recognize some of s's occurrences as named entity in all tweets of $\mathcal{T}$. Assuming there are m off-the-shelf NERs $r_1, r_2, \ldots, r_m$, we further denote f_s^R to be the number of NERs that have detected at least one occurrence of s as named entity, $f_s^R = \sum_i^m I(f_{r_i,s})$: $I(f_{r_i,s}) = 1$ if $f_{r_i,s} > 0$; $I(f_{r_i,s}) = 0$ otherwise.

We approximate the probability of s being a valid name entity (*i.e.*, a valid segment) using a voting algorithm defined by Eq. 4:

$$\hat{Pr}_{NER}(s) = w(s,m) \cdot \frac{1}{m}\sum_i^m \hat{Pr}_{r_i}(s) \quad (4)$$

$$w(s,m) = 1/\left(1 + e^{-\beta(f_s^R - m/2)}\right) \quad (5)$$

$$\hat{Pr}_{r_i}(s) = \left(1 + \frac{\alpha}{f_{r_i,s} + \epsilon}\right)^{-\frac{f_s}{f_{r_i,s}+\epsilon}} \quad (6)$$

Our approximation contains two parts. The right part of Eq. 4 (rf. Eq. 6) is the average confidence that one weak NER recognizes s as named entity. A biased estimation is simply $1/m \cdot \sum_{i=1}^m f_{r_i,s}/f_s$ because each $f_{r_i,s}/f_s$ is a noisy version of the true probability. However, such simple average ignores the absolute value of $f_{r_i,s}$ which can also play an important role here. For example, a party's name in an election event may appear hundreds of times in a tweet batch. However, due to the free writing styles of tweets, only tens of the party name's occurrences are recognized by weak NERs as named entity. In this case, $f_{r_i,s}/f_s$ is relatively small yet $f_{r_i,s}$ is relatively high. Thus, we design Eq. 6 that favors both $f_{r_i,s}/f_s$ and $f_{r_i,s}$. The favor scale is controlled by a factor α. When α is large, our function is more sensitive to the change of $f_{r_i,s}/f_s$; when α is small, a reasonably large $f_{r_i,s}$ leads $\hat{Pr}_{r_i}(s)$ to be close to 1 despite of a relatively small value of $f_{r_i,s}/f_s$. In this paper we empirically set $\alpha = 0.2$ in experiments. A small constant ϵ is set to avoid dividing by zero.

The left part of Eq. 4, $w(s,m)$ (rf. Eq. 5) uses a sigmoid function to control the impact of the majority degree of m weak NERs on the segment, which is tuned by a factor β. For example, in our paper we set $\beta = 10$ so that as long as more than half of weak NERs recognize s as named entity, $w(s,m)$ is close to 1. With a small β, $w(s,m)$ gets closer to 1 when more weak NERs recognize s as named entity.

Considering both global context and the local context by NER voting, we approximate $Pr(s)$ using a linear combination:

$$Pr(s) = (1-\lambda)Pr_{MS}(s) + \lambda\hat{Pr}_{NER}(s) \quad (7)$$

where $\hat{Pr}_{NER}(s)$ is defined by Eq. 4 with a coupling factor $\lambda \in [0,1)$, and $Pr_{MS}(\cdot)$ is the n-gram probability provided by Microsoft Web N-Gram service. The learning of λ will be detailed in Section 4.3.

4.2 Learning from Local Collocation

Collocation is defined as *an arbitrary and recurrent word combination* in [32]. Let $w_1w_2w_3$ be a valid segment, it is expected that sub-n-grams $\{w_1, w_2, w_3, w_1w_2, w_2w_3\}$ are positively correlated with one another. Thus, we need a measure that captures the extent to which the sub-n-grams of a n-gram are correlated with one another, so as to estimate the probability of the n-gram being a valid segment.

Statistical n-gram language modeling is to estimate the probability of n-gram $w_1w_2\ldots w_n$, which has been extensively studied in speech recognition and text mining [33], [34], [35], [36]. By using the chain rule, we express the n-gram probability in Eq. 8:

$$\hat{Pr}_{NGram}(w_1\ldots w_n) = \prod_{i=1}^n \hat{Pr}(w_i|w_1\ldots w_{i-1}) \quad (8)$$

where $\hat{Pr}(w_i|w_1\ldots w_{i-1})$ is the conditional probability of word w_i following word sequence $w_1\ldots w_{i-1}$. Here, we aim to quantify the strength of a n-gram being a valid segment based on the n-gram distribution in the batch of tweets. That is, we try to capture the dependencies between the sub-n-grams of a n-gram. In this sense, we set $\hat{Pr}(w_1)$ to be 1 in Eq. 8.

Absolute Discounting Smoothing. At first glance, it seems that applying maximum likelihood estimation is straightforward. However, because $Pr(w_1)$ is set to 1, then $\hat{Pr}_{NGram}(w_1 \dots w_n) = f_{w_1 \dots w_n}/f_{w_1}$. More importantly, due to the informal writing style and limited length of tweets, people often use a sub-n-gram to refer to a n-gram. For example, either first name or last name is often used in tweets to refer to the same person instead of her full name. We therefore adopt absolute discounting smoothing method [33], [34] to boost up the likelihood of a valid segment. That is, the conditional probability $Pr(w_i|w_1 \dots w_{i-1})$ is estimated by Eq. 9, where $d(w_1 \dots w_{i-1})$ is the number of distinct words following word sequence $w_1 \dots w_{i-1}$, and κ is the discounting factor.

Right-to-left Smoothing. Like most n-gram models, the model in Eq. 8 follows the writing order of left-to-right. However, it is reported that the latter words in a n-gram often carry more information [37]. For example, "justin bieber" is a bursty segment in some days of tweets data in our pilot study. Since "justin" is far more prominent than word "bieber", the n-gram probability of the segment is relative small. However, we observe that "justin" almost always precedes "bieber" when the latter occurs. Given this, we introduce a right-to-left smoothing (RLS) method mainly for name detection. Using RLS, the conditional likelihood $Pr(w_2|w_1)$ is calculated by Eq. 10, where $f_{w_1 w_2}/f_{w_2}$ is the conditional likelihood of w_1 preceding w_2, and θ is a coupling factor which balances the two parts (θ is empirically set to 0.5 in our experiments). Note that, RLS is only applied when calculating the conditional probabilities of 2-grams, because higher order n-grams have more specific information. For example, "social network" is more specific than word "social" for the estimation of the valid segment "social network analysis".

Bursty-based Weighting. Similar to that in Eq. 7, the estimation of local collocation can be combined with global context using a linear combination with a coupling factor λ:

$$Pr(s) = (1-\lambda)Pr_{MS}(s) + \lambda \hat{Pr}_{NGram}(s) \tag{11}$$

However, because tweets are noisy, the estimation of a n-gram being a valid segment is confident only when there are a lot of samples. Hence, we prefer global context in tweet segmentation when the frequency of a n-gram is relatively small. Therefore, we introduce a bursty-based weighting scheme for combining local collocation and global context.

$$Pr(s) = (1-\lambda)Pr_{MS}(s) + \lambda \mathcal{B}(s)\hat{Pr}_{NGram}(s) \tag{12}$$

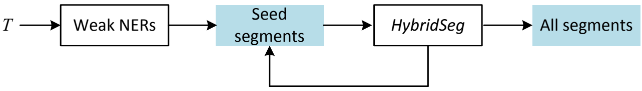


Fig. 3: The iterative process of *HybridSeg*$_{Iter}$

$\mathcal{B}(s)$, in a range of $(0,1)$, quantifies the burstiness of segment s. It satisfies two constraints: a) $\mathcal{B}(s_1) \geq \mathcal{B}(s_2)$ if $f_{s_1} \geq f_{s_2}$ and s_1 and s_2 are both i-gram segments; b) $\mathcal{B}(s_1) \geq \mathcal{B}(s_2)$ if $f_{s_1} = f_{s_2}$ and s_1 is a i-gram segment and s_2 is a j-gram segment and $i > j$. We define $\mathcal{B}(s)$ for segment s of i-gram as:

$$\mathcal{B}(s) = 1/(1 + e^{-\tau(i)(f_s - \bar{f}(i))}) \tag{13}$$

where $\bar{f}(i)$ is the average frequency of all i-grams in the batch $\mathcal{T}$, and $\tau(i)$ is a scaling function $\tau(i) = 5/\sigma(i)$, and $\sigma(i)$ is the standard deviation of the frequency of all i-grams in the batch. That is, the local collocation measure is reliable if there is enough samples of a segment in the batch.

4.3 Learning from Pseudo Feedback

As shown in Figure 2, so far in the proposed *HybridSeg* framework, each tweet is segmented independently from other tweets in a batch, though local context are derived from all tweets in the same batch. Recall that segmenting a tweet is an optimization problem. The probability of phraseness of any candidate segment in a tweet could affect its segmentation result. We therefore design an iterative process in the *HybridSeg* framework to learn from the most confident segments in the batch from the previous iteration. Figure 3 illustrates the iterative process where the confident named entities voted by weak NERs are considered as the most confident segments (or seed segments) in the 0^{th} iteration. In the subsequent iterations, the confident segments from the previous iteration become the seed segments and the same process repeats until the segmentation results of *HybridSeg* do not change significantly. We define the stop criterion using Jensen-Shannon divergence (JSD) of the frequency distributions of segments in two consecutive iterations.

Suppose at iteration i, *HybridSeg* outputs a set of segments $\{\langle s, f_s^i \rangle\}$, where f_s^i is the number of times s is a segment at iteration i. Then, f_s^i/f_s relatively records the segmentation confidence of *HybridSeg* about s at iteration i (recall that f_s denotes the frequency of s in batch $\mathcal{T}$). Similar to Eq. 6, we define

$$\hat{Pr}^{i}(s) \quad = \quad \left(1 + \frac{\alpha}{f_s^i + \epsilon}\right)^{-\frac{f_s}{f_s^i + \epsilon}}$$

Following the same combination strategy defined by Eq. 7, we have the following iterative updating function:

$$Pr^{i+1}(s) = (1-\lambda)Pr_{MS}(s) + \lambda \hat{Pr}^{i}(s), \tag{14}$$

$$\hat{Pr}(w_i|w_1\dots w_{i-1}) = \frac{\max\{f_{w_1\dots w_i}-\kappa,0\}}{f_{w_1\dots w_{i-1}}} + \frac{\kappa\cdot d(w_1\dots w_{i-1})}{f_{w_1\dots w_{i-1}}}\cdot Pr(w_i|w_2\dots w_{i-1}) \quad (9)$$

$$\hat{Pr}(w_2|w_1) = \theta\{\frac{\max\{f_{w_1w_2}-\kappa,0\}}{f_{w_1}} + \frac{\kappa\cdot d(w_1)}{f_{w_1}}\cdot Pr(w_2|w_1)\} + (1-\theta)\frac{f_{w_1w_2}}{f_{w_2}} \quad (10)$$

In the 0^{th} iteration, $\hat{Pr}^0(s)$ can be estimated based on the voting results of weak NERs or the confident n-grams learned from the batch of tweets.

Learning the parameter λ. The coupling factor λ in Eq. 14 is crucial for the convergence of *HybridSeg*. A good λ should ensure that the top confident segments from the previous iteration are detected more times in the next iteration. This is equivalent to maximizing the sum of detected frequency of the top confident segments (weighted by their stickiness scores, rf. Eq. 2) extracted from the previous iteration. Accordingly, learning the parameter λ is converted to an optimization problem as follows:

$$\begin{aligned}\hat{\lambda} &= \arg\max_{\lambda} \mu_{Iter}(\lambda) \\ &= \arg\max_{\lambda} \sum_{s\in\text{top-k at iteration } i} C^i(s)\cdot f^{i+1}(s) \quad (15)\end{aligned}$$

$C^i(s)$ is the stickiness score of s computed by *HybridSeg* in the previous iteration. Based on it, top-k segments can be retrieved. $f^{i+1}(s)$ is the detected frequency of s in the current iteration, which is an unknown function to variable λ. Therefore, the optimal λ is intractable. In our experiments, we use brute-force search strategy to find the optimal λ for each iteration and for each tweet batch. Since the update for Eq. 2 with a new λ can be easily calculated, the efficiency is not a major concern for a fixed number of λ values.

Learning λ for the 0^{th} iteration. Note that for the 0^{th} iteration, λ is learned differently because there is no segments detected from the previous iteration.

For *HybridSeg*$_{NER}$, a good λ shall ensure that the confident segments voted by m weak NERs can be detected more times in the next iteration. Let $\mathcal{N}_\cap$ be the segments that are recognized by all m NER systems (*i.e.*, $\mathcal{N}_\cap = \{s|f_s^R = m\}$). For each segment $s \in \mathcal{N}_\cap$, we consider its confident frequency to be the minimum number of times that s is recognized as named entity by one of the m NERs. Let the confident frequency of s be $f_{c,s}$, *i.e.*, $f_{c,s} = min_i^m f_{r_i,s}$. Then λ is learned as follows in the 0^{th} iteration:

$$\hat{\lambda} = \arg\max_{\lambda} \mu_{NER}(\lambda) = \arg\max_{\lambda} \sum_{s\in\mathcal{N}_\cap} \hat{Pr}^0(s)\cdot f_{c,s}\cdot f_s^0 \quad (16)$$

In this equation, $\hat{Pr}^0(s)$ is the value computed using Eq. 4; $\hat{Pr}^0(s)\cdot f_{c,s}$ serves as a weighting factor to adjust the importance of f_s^0 in learning λ. If segment s is very likely to be a named entity (*i.e.*, $\hat{Pr}^0(s)$ is high) and it has been detected many times by all NERs (*i.e.*, $f_{c,s}$ is large), then the number of times s is successfully segmented f_s^0 has a big impact on the selection of λ. On the other hand, if $\hat{Pr}^0(s)$ is low, or $f_{c,s}$ is small, or both conditions hold, then f_s^0 is less important to λ selection. By defining $f_{c,s} = min_i^m f_{r_i,s}$, Eq. 16 conservatively considers segments recognized by all weak NERs because of the noisy nature of tweets. This helps to reduce the possible oscillations resulted from different λ settings, since λ is a global factor (*i.e.*, not per-tweet dependent). On the other hand, we also assume that all the off-the-shelf NERs are reasonably good, *e.g.*, when they are applied on formal text. If there is a large number of NERs, then the definition of $f_{c,s}$ could be relaxed to reduce the impact of one or two poor-performing NERs among them.

For *HybridSeg*$_{NGram}$, because there is no initial set of confident segments, any heuristic approach may make the adaption of λ drifting away from its optimal range. Given that *HybridSeg*$_{NGram}$ exploits the local collocation based on n-gram statistical model, we argue that a common range could exist for most targeted Twitter streams. We empirically study the impact of λ to *HybridSeg*$_{NGram}$ in Section 6.

5 Segment-based Named Entity Recognition

In this paper, we select named entity recognition as a downstream application to demonstrate the benefit of tweet segmentation. We investigate two segment-based NER algorithms. The first one identifies named entities from a pool of segments (extracted by *HybridSeg*) by exploiting the co-occurrences of named entities. The second one does so based on the POS tags of the constituent words of the segments.

5.1 NER by Random Walk

The first NER algorithm is based on the observation that a named entity often co-occurs with other named entities in a batch of tweets (*i.e.*, the *gregarious* property).

Based on this observation, we build a segment graph. A node in this graph is a segment identified by *HybridSeg*. An edge exists between two nodes if they co-occur in some tweets; and the weight of the edge is measured by Jaccard Coefficient between the two corresponding segments. A random walk model is then applied to the segment graph. Let ρ_s be the stationary probability of segment s after applying random walk, the segment is then weighted by:

$$y(s) = e^{Q(s)}\cdot\rho_s \quad (17)$$

TABLE 1: Three POS tags as the indicator of a segment being a noun phrase, reproduced from [17]

Tag	Definition	Examples
N	common noun (NN, NNS)	books; someone
ˆ	proper noun (NNP, NNPS)	lebron; usa; iPad
$	numeral (CD)	2010; four; 9:30

In this equation, $e^{Q(s)}$ carries the same semantic as in Eq. 2. It indicates that a segment that frequently appears in Wikipedia as an anchor text is more likely to be a named entity. With the weighting $y(s)$, the top K segments are chosen as named entities.

5.2 NER by POS Tagger

Due to the short nature of tweets, the *gregarious* property may be weak. The second algorithm then explores the part-of-speech tags in tweets for NER by considering noun phrases as named entities using segment instead of word as a unit.

A segment may appear in different tweets and its constituent words may be assigned different POS tags in these tweets. We estimate the likelihood of a segment being a noun phrase (NP) by considering the POS tags of its constituent words of all appearances. Table 1 lists three POS tags that are considered as the indicators of a segment being a noun phrase.

Let $w^s_{i,j}$ be the j^{th} word of segment s in its i-th occurrence, we calculate the probability of segment s being an noun phrase as follow:

$$\hat{P}_{NP}(s) = \frac{\sum_i \sum_j [w^s_{i,j}]}{|s| \cdot f_s} \cdot \frac{1}{1 + e^{-5\frac{(f_s - \bar{f}_s)}{\sigma(f_s)}}} \qquad (18)$$

This equation considers two factors. The first factor estimates the probability as the percentage of the constituent words being labeled with an NP tag for all the occurrences of segment s, where $[w]$ is 1 if w is labeled as one of the three POS tags in Table 1, and 0 otherwise; For example, "chiam see tong", the name of a Singaporean politician and lawyer,[6] is labeled as ˆˆˆ (66.67%), NVV (3.70%), ˆVˆ (7.41%) and ˆVN (22.22%)[7]. By considering the types of all words in a segment, we can obtain a high probability of 0.877 for "chiam see tong". The second factor of the equation introduces a scaling factor to give more preference to frequent segments, where $\bar{f}_s$ and $\sigma(f_s)$ are the mean and standard deviation of segment frequency. The segments are then ranked by $y(s) = e^{Q(s)} \cdot \hat{P}_{NP}(s)$, *i.e.,* replacing ρ_s in Eq 17 by $\hat{P}_{NP}(s)$.

6 EXPERIMENT

We report two sets of experiments. The first set of experiments (Sections 6.1 to 6.3) aims to answer three questions: (i) does incorporating local context improve tweet segmentation quality compared to using global context alone? (ii) between learning from weak NERs and learning from local collocation, which one is more effective, and (iii) does iterative learning further improves segmentation accuracy? The second set of experiments (Section 6.4) evaluates segment-based named entity recognition.

6.1 Experiment Setting

Tweet Datasets. We used two tweet datasets in our experiments: *SIN* and *SGE*. The two datasets were used for simulating two targeted Twitter streams. The former was a stream consisting of tweets from users in a specific geographical region (*i.e.,* Singapore in this case), and the latter was a stream consisting of tweets matching some predefined keywords and hashtags for a major event (*i.e.,* Singapore General Election 2011).

We randomly selected $5,000$ tweets published on one random day in each tweet collection. Named entities were annotated by using BILOU schema [4], [14]. After discarding retweets and tweets with inconsistent annotations, $4,422$ tweets from *SIN* and $3,328$ tweets from *SGE* are used for evaluation. The agreement of annotation on tweet level is 81% and 62% for *SIN* and *SGE* respectively. The relatively low agreement for *SGE* is mainly due to the strategy of handling concepts of GRC and SMC, which refer to different types of electoral divisions in Singapore.[8] Annotators did not reach a consensus on whether a GRC/SMC should be labeled as a location name (*e.g.,* "aljunied grc" vs "aljunied"). Table 2 reports the statistics of the annotated NEs in the two datasets where f^g_s denotes the number of occurrences (or frequency) of named entity s (which is also a valid segment) in the annotated ground truth $\mathcal{G}$. Figure 4 plots the NEs' frequency distribution.

Wikipedia dump. We use the Wikipedia dump released on 30 Jan, 2010.[9] This dump contains $3,246,821$ articles and there are $4,342,732$ distinct entities appeared as anchor texts in these articles.

MS Web N-Gram. The Web N-Gram service provides access to three content types: document body, document titles and anchor texts. We use the statistics derived from document body as at April 2010.

Evaluation Metric. Recall that the task of tweet segmentation is to split a tweet into semantically meaningful segments. Ideally, a tweet segmentation method shall be evaluated by comparing its segmentation result against manually segmented tweets. However, manual segmentation of a reasonably sized data collection is extremely expensive. We choose to evaluate a tweet segmentation method based on whether the manually annotated named entities are correctly split

[6] http://en.wikipedia.org/wiki/Chiam_See_Tong

[7] V:verb including copula, auxiliaries; for example, might, gonna, ought, is, eats.

[8] http://en.wikipedia.org/wiki/Group_Representation_Constituency

[9] http://dumps.wikimedia.org/enwiki/

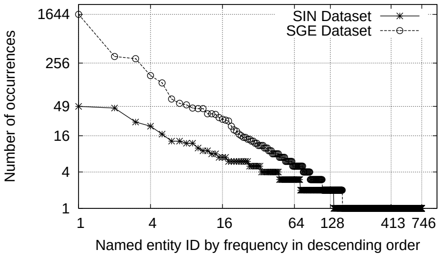


Fig. 4: Frequency distribution of the annotated NEs

TABLE 2: The annotated named entities in SIN and SGE datasets, where f_s^g denotes the frequency of named entity s in the annotated ground truth.

Dataset	#NEs	$\min f_s^g$	$\max f_s^g$	$\sum f_s^g$	#NEs *s.t.* $f_s^g > 1$
SIN	746	1	49	1234	136
SGE	413	1	1644	4073	161

as segments [1]. Because each named entity is a valid segment, the annotated named entities serve as partial ground truth in the evaluation.

We use the *Recall* measure, denoted by *Re*, which is the percentage of the manually annotated named entities that are correctly split as segments. Because a segmentation method outputs *exactly one* possible segmentation for each tweet, recall measure is the same as precision in this setting.

Methods. We evaluate 4 segmentation methods in the experiments: (i) *HybridSeg*$_{Web}$ learns from global context only, (ii) *HybridSeg*$_{NER}$ learns from global context and local context through three weak NERs, (iii) *HybridSeg*$_{NGram}$ learns from global context and local context through local collocation, and (iv) *HybridSeg*$_{Iter}$ learns from pseudo feedback iteratively on top of *HybridSeg*$_{NER}$.

The *HybridSeg*$_{NER}$ method employs three weak NERs (*i.e.*, $m = 3$) to detect named entities in tweets, namely, LBJ-NER [14], Standford-NER [15], and T-NER [3].[10] Note that, the three NERs used in our experiments are not trained using our tweets data but downloaded from their corresponding websites. The output of the three NERs over the annotated tweets are used in *HybridSeg*$_{NER}$. That is, the additional context from other unlabeled tweets published on the same day are not taken for a fair comparison.

Parameter Setting. *HybridSeg*$_{Web}$ is parameter-free. For *HybridSeg*$_{NER}$, $\alpha = 0.2$ in Eq. 6 and $\beta = 10$ in Eq. 5. The λ value in Eq. 7 is learned using an objective function in Eq. 16. Regarding parameter settings for *HybridSeg*$_{NGram}$, $\theta = 0.5$ in Eq. 10, $\kappa = 1.0$ in Eq. 9 and 10. Different values of λ in Eq. 12 are evaluated. For *HybridSeg*$_{Iter}$, the top-K segments in Eq. 15 for λ adaption is set to $K = 50$. The search space for λ is set to be $[0, 0.95]$ with a step 0.05.

TABLE 3: Recall of the 4 segmentation methods

Method	SIN	SGE
HybridSeg$_{Web}$	0.758	0.874
HybridSeg$_{NGram}$	0.806	0.907
HybridSeg$_{NER}$	0.857	0.942
HybridSeg$_{Iter}$	**0.858**	**0.946**

6.2 Segmentation Accuracy

Table 3 reports the segmentation accuracy achieved by the four methods on the two datasets. The results reported for *HybridSeg*$_{NGram}$ and *HybridSeg*$_{NER}$ are achieved with their best λ settings for fair comparison. We make three observations from the results.

(i) Both *HybridSeg*$_{NGram}$ and *HybridSeg*$_{NER}$ achieve significantly better segmentation accuracy than *HybridSeg*$_{Web}$. It shows that local context does help to improve tweet segmentation quality largely.

(ii) Learning local context through weak NERs is more effective than learning from local word collocation in improving segmentation accuracy; in particular, *HybridSeg*$_{NER}$ outperforms *HybridSeg*$_{NGram}$ on both datasets.

(iii) Iterative learning from pseudo feedback further improves the segmentation accuracy. The scale of improvement, however, is marginal. The next sub-section presents a detailed analysis of *HybridSeg* for possible reasons.

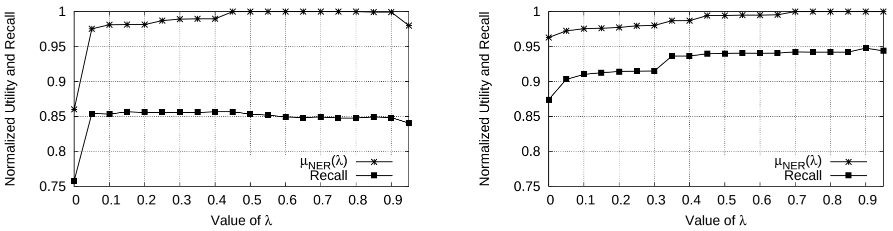
We also investigate the impact of Web N-Gram statistics for *HybridSeg*$_{Web}$ by using the other two content types: document titles and anchor texts. While the segmentation accuracy is improved up to 0.797 and 0.801 on *SIN*, the performance is degraded to 0.832 and 0.821 on *SGE*. The significant difference in performance indicates the language mismatch problem. Since the topics in *SIN* are more general [1], the specific source like document titles and anchor texts could be more discriminative for the segmentation. For the twitter streams that are topic specific like *SGE*, the language mismatch problem could become an important concern.

6.3 Method Analysis and Comparison

We first analyze and compare *HybridSeg*$_{NER}$ and *HybridSeg*$_{Ngram}$ because both learn from local context. Following this, we analyze *HybridSeg*$_{Iter}$ for the possible reasons of the marginal improvement over *HybridSeg*$_{NER}$.

***HybridSeg*$_{NER}$**. This method learns λ (rf Eq. 7) through objective function (rf Eq. 16). λ controls the combination of global and local contexts. To verify that λ can be learned through this objective function, we plot Re and $\mu_{NER}(\lambda)$ (rf Eq. 16) in Figure 5. For easy demonstration, we plot the normalized score of $\mu_{NER}(\lambda)$ in the figure. Observe that $\mu_{NER}(\lambda)$ is positively correlated with the performance metrics

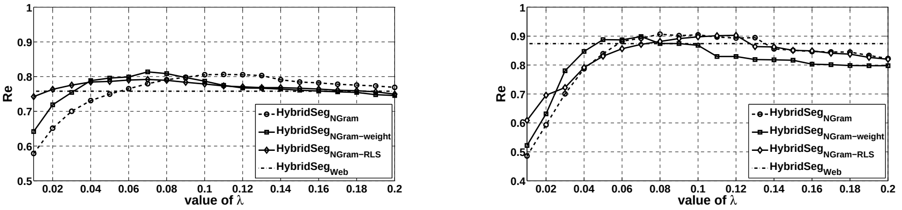
[10] Due to space constraint, readers are referred to [3], [14], [15] for details of respective NERs



(a) *SIN* dataset

(b) *SGE* dataset

Fig. 5: Re and normalized $\mu_{NER}(\lambda)$ values of *HybridSeg*$_{NER}$ with varying λ in the range of $[0, 0.95]$



(a) *SIN* dataset

(b) *SGE* dataset

Fig. 6: The impact of λ on *HybridSeg*$_{NGram}$ on the two datasets

Re on both datasets. In our experiments, we set the parameter λ to be the smallest value leading to the best $\mu_{NER}(\lambda)$, *i.e.*, $\lambda = 0.5$ on *SIN* and $\lambda = 0.7$ on *SGE*. Because λ is a global factor for all tweets in a batch and $\mu_{NER}(\lambda)$ is computed based on a small set of seed segments. A larger λ may not affect the segmentation of the seed segments because of their confident local context. But it may cause some other segments to be wrongly split due to their noisy local context. Observe there is minor degradation for Re on *SIN* dataset when $\lambda > 0.45$ although $\mu_{NER}(\lambda)$ remains the maximum.

HybridSeg$_{NGram}$. This method exploits the local collocation by using an variant of the absolute discounting based n-gram model with RLS smoothing (rf. Eq. 10) and bursty-based weighting (rf. Eq. 12). We now study the impact of the RLS smoothing and bursty-based weighting against different coupling factor λ for *HybridSeg*$_{NGram}$. Specifically, we investigate three methods with different λ settings:

- *HybridSeg*$_{NGram}$: The method with RLS smoothing and bursty-based weighting.
- *HybridSeg*$_{NGram-weight}$: The method with RLS smoothing but without bursty-based weighting.
- *HybridSeg*$_{NGram-RLS}$: The method with bursty-based weighting but without RLS smoothing.

Figure 6 reports Re of the three methods with different λ settings on both datasets. The results of *HybridSeg*$_{Web}$ is included as a baseline in the figure. Observe that with bursty-based weighting and RLS smoothing, *HybridSeg*$_{NGram}$ outperforms *HybridSeg*$_{Web}$ in a much broader range of λ values, compared to the other two alternatives. Specifically, *HybridSeg*$_{NGram}$ outperforms *HybridSeg*$_{Web}$ in the ranges of $[0.06, 0.20]$ and $[0.06, 0.13]$ on *SIN* and *SGE* datasets respectively. The figure also shows that *HybridSeg*$_{NGram}$ achieves more stable results than *HybridSeg*$_{NGram-RLS}$ and *HybridSeg*$_{NGram-weight}$ on both datasets indicating that both RLS and bursty-based weighting are helpful in achieving better segmentation results. *HybridSeg*$_{NGram}$ achieves its best performance with $\lambda \approx 0.1$ on both datasets.

HybridSeg$_{NER}$ **vs.** ***HybridSeg***$_{NGram}$. In a batch of tweets, named entities are usually a subset of the recurrent word combinations (or phrases). Therefore, *HybridSeg*$_{NGram}$ is expected to detect more segments with local context than *HybridSeg*$_{NER}$ does. However, a named entity may appear very few times in a batch. If the appearances are well formatted, there is a good chance that *HybridSeg*$_{NER}$ could detect it, but not so for *HybridSeg*$_{NGram}$ due to the limited number of appearances. As shown in the results reported earlier, *HybridSeg*$_{NER}$ does outperform *HybridSeg*$_{NGram}$.

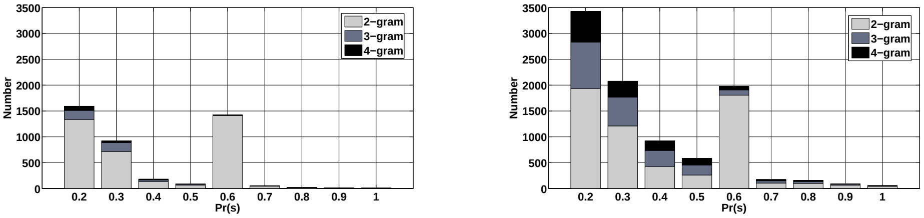
Furthermore, Table 4 lists the numbers of occurrences of the named entities that are correctly detected by *HybridSeg*$_{Web}$, *HybridSeg*$_{NER}$, and *HybridSeg*$_{NGram}$ respectively, along with the percentages of the changes relative to *HybridSeg*$_{Web}$. It shows that *HybridSeg*$_{NER}$ detects more occurrences of named entities of n-gram on both datasets when $n = 1, 2, 3, 4$. The performance of *HybridSeg*$_{NGram}$, however, is inconsistent on the two datasets.

To understand the reasons that cause inconsistent performance of *HybridSeg*$_{NGram}$ on the two datasets, we conduct a breakdown of all n-grams in terms of $\hat{Pr}_{NGram}(s)$. Figure 7 shows the distributions of $\hat{Pr}_{NGram}(s)$ of the two datasets.[11] Observe that there

[11] We ignore the n-grams whose $\hat{Pr}_{NGram}(s)$ is below 0.1

TABLE 4: Numbers of the occurrences of named entities that are correctly detected by *HybridSeg*$_{Web}$, *HybridSeg*$_{NER}$, and *HybridSeg*$_{NGram}$, and the percentage of change against *HybridSeg*$_{Web}$. #Overlap: number of the occurrences that are both detected by *HybridSeg*$_{NER}$ and *HybridSeg*$_{NGram}$.

	SIN dataset				*SGE* dataset			
n	*HybridSeg*$_{Web}$	*HybridSeg*$_{NER}$	*HybridSeg*$_{NGram}$	#Overlap	*HybridSeg*$_{Web}$	*HybridSeg*$_{NER}$	*HybridSeg*$_{NGram}$	#Overlap
1	694	793 (+14.3%)	820 (+18.2%)	767	2889	3006 (+4%)	2932 (+1.5%)	2895
2	232	246 (+6%)	172 (−25.9%)	158	519	580 (+11.8%)	600 (+15.6%)	524
3	7	12 (+71.4%)	5 (−28.6%)	4	149	238 (+59.7%)	161 (+8.1%)	143
4	2	6 (+200%)	1 (−50%)	0	1	4 (+300%)	0 (−100%)	N.A



(a) *SIN* dataset

(b) *SGE* dataset

Fig. 7: The distributions of n-grams by $Pr(s)$ for $n = 2, 3, 4$

TABLE 5: *HybridSeg*$_{Iter}$ up to 4 iterations.

Iteration	*SIN* dataset		*SGE* dataset	
	Re	*JSD*	*Re*	*JSD*
0	0.857	–	0.942	–
1	0.857	0.0059	0.946	0.0183
2	0.858	0.0001	0.946	0.0003
3	0.858	0	0.946	0

are more 2-grams in *SGE* than in *SIN* dataset that have $\hat{Pr}_{NGram}(s) > 0.5$, particularly in the range of $[0.7, 1.0]$. For $n = 3, 4$, almost no 3 or 4-grams have $\hat{Pr}_{NGram}(s) > 0.4$ on *SIN* dataset. As *SIN* contains tweets collected from a region while *SGE* is a collection of tweets on a specific topic, the tweets in *SIN* are more diverse in topics. This makes local collocation hard to capture due to their limited number of occurrences.

In summary, *HybridSeg*$_{NER}$ demonstrates more stable performance than *HybridSeg*$_{NGram}$ across different Twitter streams and achieves better accuracy. *HybridSeg*$_{NGram}$ is more sensitive to the topic specificity of Twitter streams.

Moreover, as observed in Table 4, more than 93% of the named entities detected by *HybridSeg*$_{NGram}$ are also detected by *HybridSeg*$_{NER}$. Given this, we investigate the iterative learning *HybridSeg*$_{Iter}$ on top of *HybridSeg*$_{NER}$ instead of *HybridSeg*$_{NGram}$.

Iterative Learning with *HybridSeg*$_{Iter}$. As reported in Table 3, *HybridSeg*$_{Iter}$ achieves marginal improvements over *HybridSeg*$_{NER}$. Table 5 also shows the results of *HybridSeg*$_{Iter}$ in different iterations. It is also observed that *HybridSeg*$_{Iter}$ quickly converges after two iterations. To understand the reason behind, we analyze the segments detected in each iteration. There are three categories of them:

- Fully detected segments (FS): all occurrences of the segments are detected from the batch of tweets. Their $Pr(s)$ is further increased by considering their local context. No more occurrences can be detected on this category of segments in the next iteration.
- Missed segments (MS): not a single occurrence of the segment is detected from the previous iteration. In this case, no local context information can be derived for them to increase their $Pr(s)$. They will be missed in the next iteration.
- Partially detected segments (PS): some but not all occurrences of the segments are detected. For this category of segments, local context can be derived from the detected occurrences. Depending on the local context, $Pr(s)$ will be adjusted. More occurrences may be detected or missed in the next iteration.

Table 6 reports the number of segments and their number of occurrences in each of the three sets (FS, MS, and PS). As shown in the table, very few segments are partially detected after learning from weak NERs in 0^{th} iteration (19 for *SIN* and 24 for *SGE*). The possible improvement can be made in 1^{st} iteration is to further detect the total 25 missed occurrences in *SIN* (resp. 67 in *SGE*), which accounts for 2.03% (resp. 1.64%) of all annotated NEs in the dataset. That is, the room for further performance improvement by iterative learning is marginal on both datasets.

Consider the *SIN* dataset, on average, there are about 6 detected occurrences to provide local context for each of the 19 partially detected segments. With the local context, *HybridSeg*$_{Iter}$ manages to reduce the number of partially detected segments from 19 to 11 in 1^{st} iteration and the total number of their missed instances are reduced from 25 to 14. No changes are observed for the remaining 11 partially detected segments in iteration 2. Interestingly, the number of

TABLE 6: Fully detected, missed, and partially detected segments for *HybridSeg*$_{Iter}$ (3 iterations) and *HybridSeg*$_{Web}$. #NE: number of distinct segments, #Occ: number of occurrences, #Det: number of detected occurrences, #Miss: number of missed occurrences.

Dataset	*SIN* dataset							*SGE* dataset						
Method/	Fully detected		Missed		Partially detected			Fully detected		Missed		Partially detected		
Iteration	#NE	#Occ	#NE	#Occ	#NE	#Det	#Miss	#NE	#Occ	#NE	#Occ	#NE	#Det	#Miss
0	581	944	146	152	19	113	25	295	1464	94	168	24	2374	67
1	581	959	154	163	11	98	14	291	1858	110	191	12	1996	28
2	583	961	152	161	11	98	14	289	1856	112	193	12	1996	28
HybridSeg$_{Web}$	504	647	195	214	47	113	85	234	708	140	336	40	2850	179

fully detected instances increased by 2 in 2^{nd} iteration. The best segmentation of a tweet is the one maximizes the stickiness of its member segments (rf Eq. 1). The change in the stickiness of other segments leads to the detection of these two new segments in the fully detected category, each occurs once in the dataset.

In *SGE* dataset, the 24 partially detected segments reduce to 12 in 1^{st} iteration. No more change to these 12 partially detected segments are observed in the following iteration. A manual investigation shows that the missed occurrences are wrongly detected as part of some other longer segments. For example, "NSP"[12] becomes part of "NSP Election Rally" and the latter is not annotated as a named entity. Probably because of its capitalization, "NSP Election Rally" is detected by weak NERs with strong confidence (*i.e.*, all its occurrences are detected). Due to its strong confidence, "NSP" therefore cannot be separated from this longer segment in next iteration regardless λ setting. Although "NSP Election Rally" is not annotated as a named entity, it is indeed a semantically meaningful phrase. On the other hand, a large portion of the occurrences for the 12 partially detected segments have been successfully detected from other tweets.

Compared to the baseline *HybridSeg*$_{Web}$ which does not take local context, *HybridSeg*$_{Iter}$ significantly reduces the number of missed segments, from 195 to 152 or 22% reduction on *SIN* dataset, and 20% reduction on *SGE* dataset from 140 to 112. Many of these segments are fully detected in *HybridSeg*$_{Iter}$.

TABLE 7: Accuracy of *GlobalSeg* and *HybridSeg* with RW and POS. The best results are in boldface. $*$ indicates the difference against the best F_1 is statistically significant by one-tailed paired t-test with $p < 0.01$.

Method	*SIN* dataset			*SGE* dataset		
	P	R	F_1	P	R	F_1
$Unigram_{POS}$	0.516	0.190	0.278*	0.845	0.333	0.478*
GlobalSeg$_{RW}$	0.576	0.335	0.423*	**0.929**	0.646	0.762*
HybridSeg$_{RW}$	0.618	0.343	0.441*	0.907	0.683	0.779*
GlobalSeg$_{POS}$	0.647	0.306	0.415*	0.903	0.657	0.760*
HybridSeg$_{POS}$	**0.685**	**0.352**	**0.465**	0.911	**0.686**	**0.783**

TABLE 8: Accuracy of the three weak NERs, where $*$ indicates the difference against the best F_1 is statistically significant by one-tailed paired t-test with $p < 0.01$.

Method	*SIN* dataset			*SGE* dataset		
	P	R	F_1	P	R	F_1
LBJ-NER	0.335	0.357	0.346*	0.674	0.402	0.504*
T-NER	0.273	**0.523**	0.359*	**0.696**	0.341	0.458*
Stanford-NER	**0.447**	0.448	**0.447**	0.685	**0.623**	**0.653**

6.4 Named Entity Recognition

We next evaluate the accuracy of named entity recognition based on segments. Section 5 presents two NER methods, namely random walk-based (RW-based) and POS-based NER. Through experiments, we aim to answer two questions: (i) which one of the two methods is more effective, and (ii) does better segmentation lead to better NER accuracy?

We evaluate five variations of the two methods, namely *GlobalSeg*$_{RW}$, *HybridSeg*$_{RW}$, *HybridSeg*$_{POS}$, *GlobalSeg*$_{POS}$, and $Unigram_{POS}$.[13] Here *GlobalSeg* denotes *HybridSeg*$_{Web}$ since it only uses global context, and *HybridSeg* refers to *HybridSeg*$_{Iter}$, the best method using both global and local context. The subscripts RW and POS refer to the RW-based and POS-based NER (see Section 5).

The method $Unigram_{POS}$ is the baseline which uses words (instead of segments) and POS tagging for NER. Similar to the work in [19], we extract noun phrases from the batch of tweets as named entities using regular expression. The confidence of a noun phrase is computed using a modified version of Eq. 18 by removing its first component.

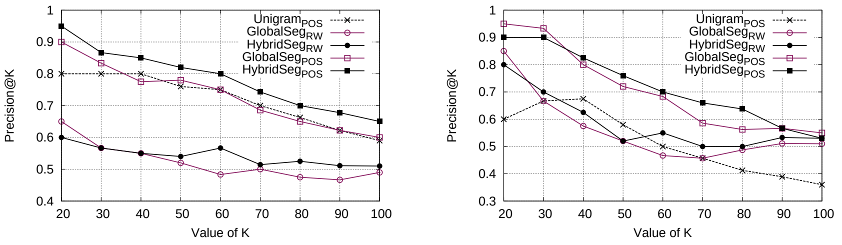
Evaluation Metric. The accuracy of NER is evaluated by Precision (P), Recall (R),[14] and F_1. P is the percentage of the recognized named entities that are truly named entities; R is the percentage of the named entities that are correctly recognized; and $F_1 = 2 \cdot P \cdot R/(P + R)$. The type of the named entity (*e.g.*, person, location, and organization) is ignored. Similar to the segmentation recall measure, each occurrence of a named entity in a specific position of a tweet is considered as one instance.

NER Results. Table 7 reports the NER accuracy of the five methods. Because all five methods are unsu-

[12] http://en.wikipedia.org/wiki/National_Solidarity_Party_(Singapore)

[13] *GlobalSeg*$_{RW}$ is the method named *TwiNER* in [1].

[14] Note R and Re are different: Re defined in Section 6.1 measures the percentage of the manually annotated named entities that are correctly split as segments.



(a) *SIN* dataset (b) *SGE* dataset

Fig. 8: Precision@K on two datasets

pervised and consider the top-K ranked segments as named entities, the results reported is the highest F_1 of each method achieved for varying $K > 50$ following the same setting in [1]. The results show that tweet segmentation greatly improves NER. $Unigram_{POS}$ is the worst performer among all methods. For a specific NER approach, either Random Walk or POS based, better segmentation results lead to better N-ER accuracy. That is, *HybridSeg*$_{RW}$ performs better than *GlobalSeg*$_{RW}$ and *HybridSeg*$_{POS}$ performs better than *GlobalSeg*$_{POS}$. Without local context in segmentation *GlobalSeg*$_{POS}$ is slightly worse than *GlobalSeg*$_{RW}$ by F_1. However, with better segmentation results, *HybridSeg*$_{POS}$ is much better than *HybridSeg*$_{RW}$. By F_1 measure, *HybridSeg*$_{POS}$ achieves the best NER result. We also observe that both the segment-based approaches *HybridSeg*$_{POS}$ and *HybridSeg*$_{RW}$ favor the popular named entities. The average frequency for correctly/wrongly recognized entities is 4.65 and 1.31 respectively based on results of $HybridSeg_{POS}$ on *SIN*. It is reasonable since the higher frequency leads to strong gregarious property for the random walk approach. Also, more instances of the named entity results in a better POS estimation for POS based approach.

For comparison, Table 8 reports the performance of the three weak NERs on the two datasets. Compared with results in Table 7, all three weak NERs perform poorly on both datasets.

Precision@K. Figure 8 plots the $Precision@K$ for the five methods on the two datasets with varying K from 20 to 100. The $Precision@K$ reports the ratio of named entities among the top-K ranked segments by each method. Note that, $Precision@K$ measures the ranking of the segments detected from a batch of tweets; the individual occurrences of each segment in the ranking are not considered. This is different from the measures (*e.g.*, Pr) reported in Table 7 where the occurrences of the named entities are considered (*i.e.*, whether a named entity is correctly detected at a specific position in a given tweet).

As observed in Figure 8, on *SIN* dataset, all methods using POS tagging for NER enjoy much better precision. RW based methods deliver much poorer precisions due to the lack of co-occurrences in the tweets. As shown in Table 2, 82% of the annotated named entities appear only once in *SIN*. Among the three POS based methods, *HybridSeg*$_{POS}$ dominates the best precisions on all K values from 20 to 100. On *SGE* dataset, the differences in precisions between POS based methods and RW based methods become smaller compared to those on *SIN* dataset. The reason is that in *SGE* dataset, about 39% of named entities appear more than once, which gives higher chance of co-occurrences. Between the two best performing methods *HybridSeg*$_{POS}$ and *GlobalSeg*$_{POS}$, the former outperforms the latter on six K values plotted between 40 and 90.

7 Conclusion

In this paper, we present the *HybridSeg* framework which segments tweets into meaningful phrases called segments using both global and local context. Through our framework, we demonstrate that local linguistic features are more reliable than term-dependency in guiding the segmentation process. This finding opens opportunities for tools developed for formal text to be applied to tweets which are believed to be much more noisy than formal text.

Tweet segmentation helps to preserve the semantic meaning of tweets, which subsequently benefits many downstream applications, e.g. named entity recognition. Through experiments, we show that segment-based named entity recognition methods achieves much better accuracy than the word-based alternative.

We identify two directions for our future research. One is to further improve the segmentation quality by considering more local factors. The other is to explore the effectiveness of the segmentation-based representation for tasks like tweets summarization, search, hashtag recommendation, etc.

References

[1] C. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, and B.-S. Lee, "Twiner: Named entity recognition in targeted twitter stream," in *SIGIR*, 2012, pp. 721–730.

[2] C. Li, A. Sun, J. Weng, and Q. He, "Exploiting hybrid contexts for tweet segmentation," in *SIGIR*, 2013, pp. 523–532.

[3] A. Ritter, S. Clark, Mausam, and O. Etzioni, "Named entity recognition in tweets: An experimental study," in *EMNLP*, 2011, pp. 1524–1534.

[4] X. Liu, S. Zhang, F. Wei, and M. Zhou, "Recognizing named entities in tweets," in *ACL*, 2011, pp. 359–367.

[5] X. Liu, X. Zhou, Z. Fu, F. Wei, and M. Zhou, "Exacting social events for tweets using a factor graph," in *AAAI*, 2012.

[6] A. Cui, M. Zhang, Y. Liu, S. Ma, and K. Zhang, "Discover breaking events with popular hashtags in twitter," in *CIKM*, 2012, pp. 1794–1798.

[7] A. Ritter, Mausam, O. Etzioni, and S. Clark, "Open domain event extraction from twitter," in *KDD*, 2012, pp. 1104–1112.

[8] X. Meng, F. Wei, X. Liu, M. Zhou, S. Li, and H. Wang, "Entity-centric topic-oriented opinion summarization in twitter," in *KDD*, 2012, pp. 379–387.

[9] Z. Luo, M. Osborne, and T. Wang, "Opinion retrieval in twitter," in *ICWSM*, 2012.

[10] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang, "Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach," in *CIKM*, 2011, pp. 1031–1040.

[11] K.-L. Liu, W.-J. Li, and M. Guo, "Emoticon smoothed language models for twitter sentiment analysis," in *AAAI*, 2012.

[12] S. Hosseini, S. Unankard, X. Zhou, and S. W. Sadiq, "Location oriented phrase detection in microblogs," in *DASFAA*, 2014, pp. 495–509.

[13] C. Li, A. Sun, and A. Datta, "Twevent: segment-based event detection from tweets," in *CIKM*, 2012, pp. 155–164.

[14] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *CoNLL*, 2009, pp. 147–155.

[15] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *ACL*, 2005, pp. 363–370.

[16] G. Zhou and J. Su, "Named entity recognition using an hmm-based chunk tagger," in *ACL*, 2002, pp. 473–480.

[17] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith, "Part-of-speech tagging for twitter: annotation, features, and experiments," in *ACL-HLT*, 2011, pp. 42–47.

[18] B. Han and T. Baldwin, "Lexical normalisation of short text messages: Makn sens a #twitter," in *ACL*, 2011, pp. 368–378.

[19] F. C. T. Chua, W. W. Cohen, J. Betteridge, and E.-P. Lim, "Community-based classification of noun phrases in twitter," in *CIKM*, 2012, pp. 1702–1706.

[20] S. Cucerzan, "Large-scale named entity disambiguation based on wikipedia data," in *EMNLP-CoNLL*, 2007, pp. 708–716.

[21] D. N. Milne and I. H. Witten, "Learning to link with wikipedia," in *CIKM*, 2008, pp. 509–518.

[22] S. Guo, M.-W. Chang, and E. Kiciman, "To link or not to link? a study on end-to-end tweet entity linking," in *HLT-NAACL*, 2013, pp. 1020–1030.

[23] A. Sil and A. Yates, "Re-ranking for joint named-entity recognition and linking," in *CIKM*, 2013, pp. 2369–2374.

[24] J. Gao, M. Li, C. Huang, and A. Wu, "Chinese word segmentation and named entity recognition: A pragmatic approach," in *Comput. Linguist.*, 2005, pp. 531–574.

[25] Y. Zhang and S. Clark, "A fast decoder for joint word segmentation and pos-tagging using a single discriminative model," in *EMNLP*, 2010, pp. 843–852.

[26] W. Jiang, L. Huang, and Q. Liu, "Automatic adaption of annotation standards: Chinese word segmentation and pos tagging - a case study," in *ACL*, 2009, pp. 522–530.

[27] X. Zeng, D. F. Wong, L. S. Chao, and I. Trancoso, "Graph-based semi-supervised model for joint chinese word segmentation and part-of-speech tagging," in *ACL*, 2013, pp. 770–779.

[28] W. Jiang, M. Sun, Y. Lü, Y. Yang, and Q. Liu, "Discriminative learning with natural annotations: Word segmentation as a case study," in *ACL*, 2013, pp. 761–769.

[29] R. Mihalcea and A. Csomai, "Wikify!: linking documents to encyclopedic knowledge," in *CIKM*, 2007, pp. 233–242.

[30] W. Wu, H. Li, H. Wang, and K. Q. Zhu, "Probase: a probabilistic taxonomy for text understanding," in *SIGMOD*, 2012, pp. 481–492.

[31] K. Wang, C. Thrasher, E. Viegas, X. Li, and P. Hsu, "An overview of microsoft web n-gram corpus and applications," in *HLT-NAACL*, 2010, pp. 45–48.

[32] F. A. Smadja, "Retrieving collocations from text: Xtract," *Comput. Linguist.*, vol. 19, no. 1, pp. 143–177, 1993.

[33] H. Ney, U. Essen, and R. Kneser, "On structuring probabilistic dependences in stochastic language modelling," *Computer Speech & Language*, vol. 8, pp. 1 – 38, 1994.

[34] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proc. of ACL*, 1996, pp. 310–318.

[35] F. Peng, D. Schuurmans, and S. Wang, "Augmenting naive bayes classifiers with statistical language models," *Inf. Retr.*, vol. 7, pp. 317–345, 2004.

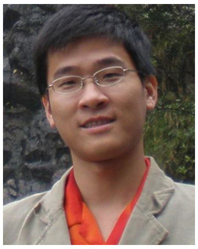
[36] K. Nishida, T. Hoshide, and K. Fujimura, "Improving tweet stream classification by detecting changes in word probability," in *Proc. of SIGIR*, 2012, pp. 971–980.

[37] X. Wang, A. McCallum, and X. Wei, "Topical n-grams: Phrase and topic discovery, with an application to information retrieval," in *ICDM*, 2007, pp. 697–702.

Chenliang Li is an Associate Professor at State Key Laboratory of Software Engineering, Computer School, Wuhan University, China. He received PhD from Nanyang Technological University, Singapore, in 2013. His research interests include information retrieval, text/web mining, data mining and natural language processing. His papers appear in SIGIR, CIKM and JASIST.

Aixin Sun is an Associate Professor with School of Computer Engineering, Nanyang Technological University, Singapore. He received PhD from the same school in 2004. His research interests include information retrieval, text mining, social computing, and multimedia. His papers appear in major international conferences like SIGIR, KDD, WSDM, ACM Multimedia, and journals including DMKD, TKDE, and JASIST.

Jianshu Weng received PhD from Nanyang Technological University, Singapore, in 2008. He is a consultant with Accenture Analytics Innovation Center, Singapore. Before joining Accenture, he was a researcher with HP Labs, Singapore. His research interests include social network analysis, text mining, opinion mining, collaborative filtering and trust-aware resource selection.



Qi He is a Senior Researcher at Relevance Science, LinkedIn. After receiving the Ph.D. from Nanyang Technological University in 2008, he conducted 2-year postdoctoral research at Pennsylvania State University for CiteSeerX, followed by a Research Staff Member position at IBM Almaden Research Center till May 2013. His research interests cover Information Retrieval/Extraction, Recommender Systems, Social Network Analysis, Data Mining and Machine Learning. He served as general co-chair of ACM CIKM 2013. He received IBM Almaden Service Research Best Paper Award and IBM Invention Achievement Award in 2012, Best Application Paper Award at SIGKDD 2008, Microsoft Research Fellowship in 2006. He is a member of ACM and IEEE.