QuickView: NLP-based Tweet Search

Xiaohua Liu [‡ †], **Furu Wei** [†], **Ming Zhou** [†], **Microsoft QuickView Team** [†]
[‡]School of Computer Science and Technology
Harbin Institute of Technology, Harbin, 150001, China
[†]Microsoft Research Asia
Beijing, 100190, China
[†]{xiaoliu, fuwei, mingzhou,qv}@microsoft.com

Abstract

Tweets have become a comprehensive repository for real-time information. However, it is often hard for users to quickly get information they are interested in from tweets, owing to the sheer volume of tweets as well as their noisy and informal nature. We present *QuickView*, an NLP-based tweet search platform to tackle this issue. Specifically, it exploits a series of natural language processing technologies, such as tweet normalization, named entity recognition, semantic role labeling, sentiment analysis, tweet classification, to extract useful information, i.e., named entities, events, opinions, etc., from a large volume of tweets. Then, non-noisy tweets, together with the mined information, are indexed, on top of which two brand new scenarios are enabled, i.e., categorized browsing and advanced search, allowing users to effectively access either the tweets or fine-grained information they are interested in.

1 Introduction

Tweets represent a comprehensive fresh information repository. However, users often have difficulty finding information they are interested in from tweets, because of the huge number of tweets as well as their noisy and informal nature. Tweet search, e.g., Twitter [1], is a kind of service aiming to tackle this issue. Nevertheless, existing tweet search services provide limited functionality. For example, in Twitter, only a simple keyword-based search is supported, and the returned list often contains meaningless results.

This demonstration introduces *QuickView*, which employs a series of NLP technologies to extract useful information from a large volume of tweets. Specifically, for each tweet, it first conducts normalization, followed by named entity recognition (NER). Then it conducts semantic role labeling (SRL) to get predicate-argument structures, which are further converted into events, i.e., triples of who did what. After that, it performs sentiment analysis (SA), i.e., extracting positive or negative comments about something/somebody. Next, tweets are classified into predefined categories. Finally, non-noisy tweets together with the mined information are indexed.

On top of the index, *QuickView* enables two brand new scenarios, allowing users to effectively access the tweets or fine-grained information mined from tweets.

Categorized Browsing. As illustrated in Figure 1(a), *QuickView* shows recent popular tweets, entities, events, opinions and so on, which are organized by categories. It also extracts and classifies URL links in tweets and allows users to check out popular links in a categorized way.

Advanced Search. As shown in Figure 1(b), *QuickView* provides four advanced search functions: 1) search results are clustered so that tweets about the same/similar topic are grouped together, and for each cluster only the informative tweets are kept; 2) when the query refers to a person or a company, two bars are presented followed by the words that strongly suggest opinion polarity. The bar's width

[1] http://twitter.com/

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 13–18,
Jeju, Republic of Korea, 8-14 July 2012.

is proportional to the number of associated opinions; 3) similarly, the top six most frequent words that most clearly express event occurrences are presented; 4) users can search tweets with opinions or events, e.g., search tweets containing any positive/negative opinion about "Obama" or any event involving "Obama".

The implementation of *QuickView* requires adapting existing NLP components trained on formal texts, which often performs poorly on tweets. For example, the average F1 of the Stanford NER (Finkel et al., 2005) drops from 90.8% (Ratinov and Roth, 2009) to 45.8% on tweets, while Liu et al. (2010) report that the F1 score of a state-of-the-art SRL system (Meza-Ruiz and Riedel, 2009) falls to 42.5% on tweets as apposed to 75.5% on news. However, the adaptation of those components is challenging, owing to the lack of annotated tweets and the inadequate signals provided by a noisy and short tweet. Our general strategy is to leverage existing resources as well as unsupervised or semi-supervised learning methods to reduce the labeling efforts, and to aggregate as much evidence as possible from a broader context to compensate for the lack of information in a tweet.

This strategy is embodied by various components we have developed. For example, our NER component combines a k-nearest neighbors (KNN) classifier, which collects global information across recently labeled tweets with a Conditional Random Fields (CRF) labeler, which exploits information from a single tweet and the gazetteers. Both the KNN classifier and the CRF labeler are repeatedly retrained using the results that they have confidently labeled. The SRL component caches and clusters recent labeled tweets, and aggregates information from the cluster containing the tweet. Similarly, the classifier considers not only the current tweet but also its neighbors in a tweet graph, where two tweets are connected if they are similar in content or have a tweet/retweet relationship.

QuickView has been internally deployed, and received extremely positive feedback. Experimental results on a human annotated dataset also indicate the effectiveness of our adaptation strategy.

Our contributions are summarized as follows.

1. We demonstrate *QuickView*, an NLP-based tweet search. Different from existing methods, it exploits a series of NLP technologies to extract useful information from a large volume of tweets, and enables categorized browsing and advanced search scenarios, allowing users to efficiently access information they are interested in from tweets.

2. We present core components of *QuickView*, focusing on how to leverage existing resources and technologies as well as how to make up for the limited information in a short and often noisy tweet by aggregating information from a broader context.

The rest of this paper is organized as follows. In the next section, we introduce related work. In Section 3, we describe our system. In Section 4, we evaluate our system. Finally, Section 5 concludes and presents future work.

2 Related Work

Information Extraction Systems. Essentially, *QuickView* is an information extraction (IE) system. However, unlike existing IE systems, such as Evita (Saurí et al., 2005), a robust event recognizer for QA system, and SRES (Rozenfeld and Feldman, 2008), a self-supervised relation extractor for the web, it targets tweets, a new genre of text, which are short and informal, and its focus is on adapting existing IE components to tweets.

Tweet Search Services. A couple of tweet search services exist, including Twitter, Bing social search [2] and Google social search [3]. Most of them provide only keyword-based search interfaces, i.e., returning a list of tweets related to a given word/phrase. In contrast, our system extracts fine-grained information from tweets and allows a new end-to-end search experience beyond keyword search, such as clustering of search results, and search with events/opinions.

NLP Components. The NLP technologies adopted in our system , e.g., NER, SRL and classification, have been extensively studied on formal text but rarely on tweets. At the heart of our system is the re-use of existing resources, methodologies as

[2] http://www.bing.com/social

[3] http://www.google.com/realtime

(a) A screenshot of the categorized browsing scenario.

(b) A screenshot of the advanced search scenario.

Figure 1: Two scenarios of *QuickView*.

well as components, and the the adaptation of them to tweets. The adaptation process, though varying across components, consists of three common steps: 1) annotating tweets; 2) defining the decision context that usually involves more than one tweet, such as a cluster of similar tweets; and 3) re-training models (often incrementally) with both conventional features and features derived from the context defined in step 2.

3 System Description

We first give an overview of our system, then present more details about NER and SRL, as two representative core components, to illustrate the adaptation process.

3.1 Overview

Architecture. *QuickView* can be divided into four parts, as illustrated in Figure 2. The first part includes a crawler and a buffer of raw tweets. The crawler repeatedly downloads tweets using the Twitter APIs, and then pre-filters noisy tweets using some heuristic rules, e.g., removing a tweet if it is too short, say, less than 3 words, or if it contains any predefined banned word. At the moment, we focus on English tweets, so non-English tweets are filtered as well. Finally, the un-filtered are put into the buffer.

The second part consists of several tweet extraction pipelines. Each pipeline has the same configuration, constantly fetching a tweet from the raw tweet buffer, and conducting the following processes se-

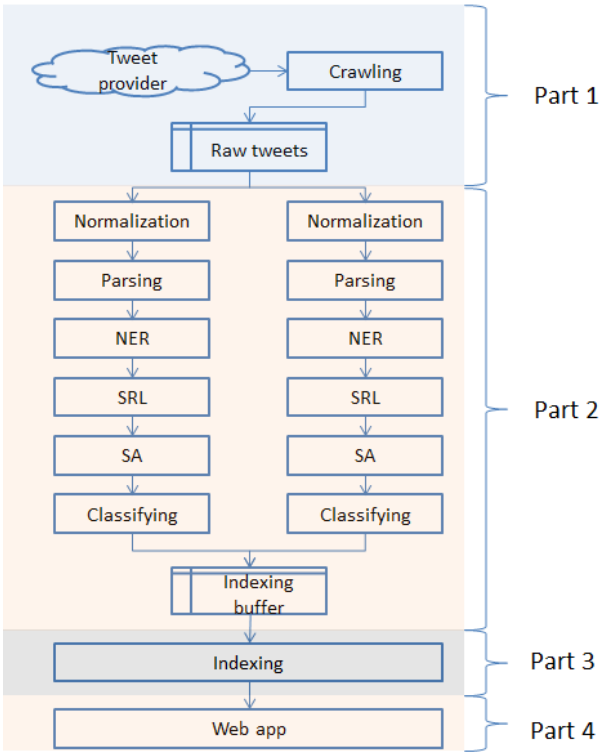

Figure 2: System architecture of *QuickView*.

quentially: 1) normalization; 2) parsing including part-of-speech (POS), chunking, and dependency parsing; 3) NER; 4) SRL; 5) SA and 6) classification. The normalization model identifies and corrects ill-formed words. For example, after normalization, "looooove" in "$\cdots$I looooove my icon$\cdots$" will be transformed to "love". A phrase-based translation system without re-ordering is used to implement this model. The translation table includes manually compiled ill/good form pairs, and the language model is a trigram trained on LDC data [4] using SRILM (Stolcke, 2002). The OpenNLP [5] toolkit is directly used to implement the parsing model. In future, the parsing model will be re-trained using annotated tweets. The SA component is implemented according to Jiang et al. (2011), which incorporates target-dependent features and considers related tweets by utilizing a graph-based optimization. The classification model is a KNN-based classifier that caches confidently labeled results to re-train itself, which also recognizes and drops noisy tweets. Each processed tweet, if not identified as noise, is put into a shared buffer for indexing.

The third part is responsible for indexing and querying. It constantly takes from the indexing buffer a processed tweet, which is then indexed with various entries including words, phrases, metadata (e.g., source, publish time, and account), named entities, events, and opinions. On top of this, it answers any search request, and returns a list of matched results, each of which contains both the original tweet and the extracted information from that tweet. We implement an indexing/querying engine similar to Lucene [6] in C#. This part also maintains a cache of recent processed tweets, from which the following information is extracted and indexed: 1) top tweets; 2) top entities/events/opinions in tweets; and 3) top accounts. Whether a tweet/entity/event/opinion ranks top depends on their re-tweeted/mentioned times as well as its publisher, while whether an account is top relies on the number of his/her followers and tweets.

The fourth part is a web application that returns related information to end users according to their browsing or search request. The implementation of the web application is organized with the model-view-control pattern so that other kinds of user interfaces, e.g., a mobile application, can be easily implemented.

Deployment. *QuickView* is deployed into 5 workstations [7] including 2 processing pipelines, as illustrated in Table 1. The communication between components is through TCP/IP. On average, it takes 0.01 seconds to process each tweet, and in total about 10 million tweets are indexed every day. Note that *QuickView*'s processing capability can be enhanced in a straightforward manner by deploying additional pipelines.

3.2 Core Components

Because of limited space, we only discuss two core components of *QuickView*: NER and SRL.

NER. NER is the task of identifying mentions of rigid designators from text belonging to named-entity types such as persons, organizations and locations. Existing solutions fall into three categories: 1)

[4] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp ?catalogId=LDC2005T12

[5] http://sourceforge.net/projects/opennlp/

[6] http://lucene.apache.org/java/docs/index.html

[7] Intel® Xeon® 2.33 CPU 5140 @2.33GHz, 4G of RAM, OS of Windows Server 2003 Enterprise $X64$ version

Table 1: Current deployment of *QuickView*.

Workstation	Hosted components
$\#1$	Crawler,Raw tweet buffer
$\#2, 3$	Process pipeline
$\#4$	Indexing Buffer, Indexer/Querier
$\#5$	Web application

the rule-based (Krupka and Hausman, 1998); 2) the machine learning based (Finkel and Manning, 2009; Singh et al., 2010); and 3) hybrid methods (Jansche and Abney, 2002). With the availability of annotated corpora, such as ACE05, Enron and CoNLL03, the data-driven methods become the dominating methods. However, because of domain mismatch, current systems trained on non-tweets perform poorly on tweets.

Our NER system takes three steps to address this problem. Firstly, it defines those recently labeled tweets that are similar to the current tweet as its recognition context, under which a KNN-based classifier is used to conduct word level classification. Following the two-stage prediction aggregation methods (Krishnan and Manning, 2006), such pre-labeled results, together with other conventional features used by the state-of-the-art NER systems, are fed into a linear CRF models, which conducts fine-grained tweet level NER. Secondly, the KNN and CRF model are repeatedly retrained with an incrementally augmented training set, into which highly confidently labeled tweets are added. Finally, following Lev Ratinov and Dan Roth (2009), 30 gazetteers are used, which cover common names, countries, locations, temporal expressions, etc. These gazetteers represent general knowledge across domains, and help to make up for the lack of training data.

SRL. Given a sentence, the SRL component identifies every predicate, and for each predicate further identifies its arguments. This task has been extensively studied on well-written corpora like news, and a couple of solutions exist. Examples include: 1) the pipelined approach, i.e., dividing the task into several successive components such as argument identification, argument classification, global inference, etc., and conquering them individually (Xue, 2004; Koomen et al., 2005); 2) sequentially labeling based approach (Màrquez et al., 2005), i.e., labeling the words according to their positions relative to an argument (i.e., inside, outside, or at the beginning); and 3) Markov Logic Networks (MLN) based approach (Meza-Ruiz and Riedel, 2009), i.e., simultaneously resolving all the sub-tasks using learnt weighted formulas. Unsurprisingly, the performance of the state-of-the-art SRL system (Meza-Ruiz and Riedel, 2009) drops sharply when applied to tweets.

The SRL component of *QuickView* is based on CRF, and uses the recently labeled tweets that are similar to the current tweet as the broader context. Algorithm 1 outlines its implementation, where: $train$ denotes a machine learning process to get a labeler l, which in our work is a linear CRF model; the $cluster$ function puts the new tweet into a cluster; the $label$ function generates predicate-argument structures for the input tweet with the help of the trained model and the cluster; p, s and cf denote a predicate, a set of argument and role pairs related to the predicate and the predicted confidence, respectively. To prepare the initial clusters required by the SRL component as its input, we adopt the predicate-argument mapping method (Liu et al., 2010) to get some automatically labeled tweets, which (plus the manually labeled tweets) are then organized into groups using a bottom-up clustering procedure.

It is worth noting that: 1) our SRL component uses the general role schema defined by PropBank, which includes core roles such as A0, A1 (usually indicating the agent and patient of the predicate, respectively), and auxiliary roles such as AM-TMP and AM-LOC (representing the temporal and location information of the predicate, respectively); 2) only verbal predicates are considered, which is consistent with most existing SRL systems; and 3) following Màrquez et al. (2005), it conducts word level labeling.

4 Evaluation

Overall Performance. We provide a textbox in the home page of *QuickView* to collect feedback. We have got 165 feedbacks, of which 85.5% are positive. The main complaint is related to the quality of the extracted information.

Core Components. We manually labeled the POS,

Algorithm 1 SRL of *QuickView*.

Require: Tweet stream i;clusters cl;output stream o.

1: Initialize l, the CRF labeler: $l = train(cl)$.
2: **while** Pop a tweet t from i and $t \neq null$ **do**
3: Put t to a cluster c: $c = cluster(cl, t)$.
4: Label t with l:$(t, \{(p, s, cf)\}) = label(l, c, t)$.
5: Update cluster c with labeled results $(t, \{(p, s, cf)\})$.
6: Output labeled results $(t, \{(p, s, cf)\})$ to o.
7: **end while**
8: **return** o.

NER, SRL and SA information for about 10,000 tweets, based on which the NER and SRL components are evaluated. Experimental results show that: 1) our NER component achieves an average F1 of 80.2%, as opposed to 75.4% of the baseline, which is a CRF-based system similar to Ratinov and Roth's (2009) but re-trained on annotated tweets; and 2) our SRL component gets an F1 of 59.7%, outperforming both the state-of-the-art system (Meza-Ruiz and Riedel, 2009) (42.5%) and the system of Liu et al. (2010) (42.3%), which is trained on automatically annotated news tweets (tweets reporting news).

5 Conclusions and Future work

We have described the motivation, scenarios, architecture, deployment and implementation of *QuickView*, an NLP-based tweet search. At the heart of *QuickView* is the adaptation of existing NLP technologies, e.g., NER, SRL and SA, to tweets, a new genre of text, which are short and informal. We have illustrated our strategy to tackle this challenging task, i.e., leveraging existing resources and aggregating as much information as possible from a broader context, using NER and SRL as case studies. Preliminary positive feedback suggests the usefulness of *QuickView* and its advantages over existing tweet search services. Experimental results on a human annotated dataset indicate the effectiveness of our adaptation strategy.

We are improving the quality of the core components of *QuickView* by labeling more tweets and exploring alternative models. We are also customizing *QuickView* for non-English tweets. As it progresses, we will release *QuickView* to the public.

References

Jenny Rose Finkel and Christopher D. Manning. 2009. Nested named entity recognition. In *EMNLP*, pages 141–150.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, pages 363–370.

Martin Jansche and Steven P. Abney. 2002. Information extraction from voicemail transcripts. In *EMNLP*, pages 320–327.

Long Jiang, Mo Yu, Ming Zhou, and Xiaohua Liu. 2011. Target-dependent twitter sentiment classification. In *ACL*.

Peter Koomen, Vasin Punyakanok, Dan Roth, and Wentau Yih. 2005. Generalized inference with multiple semantic role labeling systems. In *CONLL*, pages 181–184.

Vijay Krishnan and Christopher D. Manning. 2006. An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *ACL*, pages 1121–1128.

George R. Krupka and Kevin Hausman. 1998. Isoquest: Description of the netowl^{TM} extractor system as used in muc-7. In *MUC-7*.

Xiaohua Liu, Kuan Li, Bo Han, Ming Zhou, Long Jiang, Zhongyang Xiong, and Changning Huang. 2010. Semantic role labeling for news tweets. In *Coling*, pages 698–706.

Lluís Màrquez, Pere Comas, Jesús Giménez, and Neus Català. 2005. Semantic role labeling as sequential tagging. In *CONLL*, pages 193–196.

Ivan Meza-Ruiz and Sebastian Riedel. 2009. Jointly identifying predicates, arguments and senses using markov logic. In *NAACL*, pages 155–163.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *CoNLL*, pages 147–155.

Benjamin Rozenfeld and Ronen Feldman. 2008. Self-supervised relation extraction from the web. *Knowl. Inf. Syst.*, 17:17–33, October.

Roser Saurí, Robert Knippen, Marc Verhagen, and James Pustejovsky. 2005. Evita: A robust event recognizer for qa systems. In *EMNLP*, pages 700–707.

Sameer Singh, Dustin Hillard, and Chris Leggetter. 2010. Minimally-supervised extraction of entities from text advertisements. In *HLT-NAACL*, pages 73–81.

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *ICSLP*, volume 2, pages 901–904.

Nianwen Xue. 2004. Calibrating features for semantic role labeling. In *In Proceedings of EMNLP 2004*, pages 88–94.