International Journal of Research
Available
https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 14
October 2016

Application of Named Entity wise Recognition and Tweet Segmentation

N. HAVYA
DEPARTMENT OF CSE
MADHIRA INSTITUTE OF TECHNOLOGY AND SCIENCES

Mr. B.NARASIMHA RAO, M. Tech
Assistant Professor
DEPARTMENT OF CSE
MADHIRA INSTITUTE OF TECHNOLOGY AND SCIENCES

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 stickinessscores 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 data sets 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.

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) event detection and summarization opinion mining sentiment analysis 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 January 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 consecutiven-grams (n _ 1Þ, 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]. Fig. 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 no 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 extraction 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

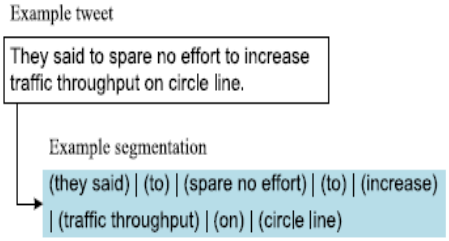

Fig. 1. Example of tweet segmentation.

named entity. In [6] the segment "korea versus 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 HybridSegWeb.

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 HybridSegNER. It obtains confident segments based on the voting results of multiple off-the-shelf NER tools. Another method utilizing local collocation knowledge, denoted by HybridSegNGram, is proposed based on the observation that many tweets published within a short time period are about the same topic. HybridSegNGram segments tweets by estimating the term-dependency within a batch of tweets.

4.3.3 IMPLEMENTATION

Admin

In this module, the Admin has to login by using valid user name and password. After login successful he can do some operations such as search history, view users, request & response, all topic messages and topics.

Search History

This is controlled by admin; the admin can view the search history details. If he clicks on search history button, it will show the list of searched user details with their tags such as user name, searched user, time and date.

Request & Response

In this module, the admin can view the all the friend request and response. Here all the request and response will be stored with their tags such as Id, requested user photo, requested user name, user name request to, status and time & date. If the user accepts the request then status is accepted or else the status is waiting.

Tweet segmentation Topic Messages

In this module, the admin can view the messages such as emerging topic messages and Anomaly emerging topic messages. Tweet segmentation topic messages means we can send a message to particular user.

User

In this module, there are n numbers of users are present. User should register before doing some operations. And register user details are stored in user module. After registration successful he has to login by using authorized user name and password. Login successful he will do some operations like view or search users, send friend request, view messages, send messages, Tweet segmentation messages and followers.

Search Users

The user can search the users based on users and the server will give response to the user like User name, user image, E mail id, phone number and date of birth. If you want send friend request to particular receiver then click on follow, then request will send to the user.

Messages

User can view the messages, send messages and send anomaly messages to users. User can send messages based on topic to the particular user, after sending a message that topic rank will be increased. Then again another user will also re-tweet the particular topic then that topic rank will increases. The anomaly message means user wants send a message to all users.

Followers

In this module, we can view the followers' details with their tags such as user name, user image, date of birth, E mail ID, phone number and ranks.

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 andom walk model is then applied to the segment graph. Let rs be the stationary probability of segment s after applying random walk, the segment is then weighted by yðsÞ ¼ eQðsÞ _ rs: (17) In this equation, eQðsÞ carries the same semantic as in 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 yðsÞ, the top K segments are chosen as
named entities.

SYSTEM DESIGN AND DEVELOPMENT

INPUT DESIGN

Input Design plays a vital role in the life cycle of software development, it requires very careful attention of developers. The input design is to feed data to the application as accurate as possible. So inputs are supposed to be designed effectively so that the errors occurring while feeding are minimized. According to Software Engineering Concepts, the input forms or screens are designed to provide to have a validation control over the input limit, range and other related validations.

This system has input screens in almost all the modules. Error messages are developed to alert the user whenever he commits some mistakes and guides him in the right way so that invalid entries are not made. Let us see deeply about this under module design. Input design is the process of converting the user created input into a computer-based format. The goal of the input design is to make the data entry logical and free from errors. The error is in the input are controlled by the input design. The application has been developed in user-friendly manner. The forms have been designed in such a way during the processing the cursor is placed in the position where must be entered. The user is also provided with in an option to select an appropriate input from various alternatives related to the field in certain cases. Validations are required for each data entered. Whenever a user enters an erroneous data, error message is displayed and the user can move on to the subsequent pages after completing all the entries in the current page

OUTPUT DESIGN

The Output from the computer is required to mainly create an efficient method of communication within the company primarily among the project leader and his team members, in other words, the administrator and the clients. The output of VPN is the system which allows the project leader to manage his clients in terms of creating new clients and assigning new projects to them, maintaining a record of the project validity and providing folder level access to each client on the user side depending on the projects allotted to him. After completion of a project, a new project may be assigned to the client. User authentication procedures are maintained at the initial stages itself. A new user may be created by the

administrator himself or a user can himself register as a new user but the task of assigning projects and validating a new user rests with the administrator only. The application starts running when it is executed for the first time. The server has to be started and then the internet explorer in used as the browser. The project will run on the local area network so the server machine will serve as the administrator while the other connected systems can act as the clients. The developed system is highly user friendly and can be easily understood by anyone using it even for the first time.

SYSTEM TESTING

TESTING METHODOLOGIES

The following are the Testing Methodologies:

Unit Testing.

Integration Testing.

User Acceptance Testing.

Output Testing.

Validation Testing.

Unit Testing

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to

ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

Integration Testing

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The following are the types of Integration Testing:

1. Top Down Integration

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

2. Bottom-up Integration

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.

A driver (i.e.) the control program for testing is written to coordinate test case input and output.

The cluster is tested.

Drivers are removed and clusters are combined moving upward in the program structure

The bottom up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

User Acceptance Testing

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce

the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

Validation Checking

Validation checks are performed on the following fields.

Text Field:

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

Numeric Field:

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

Preparation of Test Data

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

Using Live Test Data:

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

Using Artificial Test Data:

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications. The package “Virtual Private Network” has satisfied all the requirements specified as per software requirement specification and was accepted.

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 Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2012, pp. 721–730.
[2] C. Li, A. Sun, J. Weng, and Q. He, "Exploiting hybrid contexts for tweet segmentation," in Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2013, pp. 523–532.
[3] A. Ritter, S. Clark, Mausam, and O. Etzioni, "Named entity recognition in tweets: An experimental study," in Proc. Conf. Empirical Methods Natural Language Process., 2011, pp. 1524–1534.
[4] X. Liu, S. Zhang, F. Wei, and M. Zhou, "Recognizing named entities in tweets," in Proc. 49th Annu. Meeting Assoc. Comput. Linguistics: Human Language Technol., 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 Proc. AAAI Conf. Artif. Intell., 2012, pp. 1692–1698.
[6] A. Cui, M. Zhang, Y. Liu, S. Ma, and K. Zhang, "Discover breaking events with popular hashtags in twitter," in Proc. 21st ACM Int. Conf. Inf. Knowl. Manage., 2012, pp. 1794–1798.
[7] A. Ritter, Mausam, O. Etzioni, and S. Clark, "Open domain event extraction from twitter," in Proc. 18th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining, 2012, pp. 1104–1112.

[8] X. Meng, F. Wei, X. Liu, M. Zhou, S. Li, and H. Wang, "Entitycentric topic-oriented opinion summarization in twitter," in Proc. 18th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining, 2012, pp. 379–387.

[9] Z. Luo, M. Osborne, and T. Wang, "Opinion retrieval in twitter," in Proc. Int. AAAI Conf. Weblogs Social Media, 2012, pp. 507–510.
[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 Proc. 20th ACM Int. Conf. Inf. Knowl. Manage., 2011, pp. 1031–1040.
[11] K.-L. Liu, W.-J. Li, and M. Guo, "Emoticon smoothed language models for twitter sentiment analysis," in Proc. AAAI Conf. Artif. Intell., 2012, pp. 1678–1684.
[12] S. Hosseini, S. Unankard, X. Zhou, and S. W. Sadiq, "Location oriented phrase detection in microblogs," in Proc. 19th Int. Conf. Database Syst. Adv. Appl., 2014, pp. 495–509.
[13] C. Li, A. Sun, and A. Datta, "Twevent: segment-based event detection from tweets," in Proc. 21st ACM Int. Conf. Inf. Knowl. Manage., 2012, pp. 155–164.
[14] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in Proc. 13th Conf. Comput. Natural Language Learn., 2009, pp. 147–155.
[15] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating nonlocal information into information extraction systems by Gibbs sampling," in Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics, 2005, pp. 363–370.