“Where can I Buy iPhone4S Now?”: Spatio-Temporal Entity Retrieval on Twitter

Liang YAN, Qiang MA, Masatoshi YOSHIKAWA

Department of Social Informatics, Graduate School of Informatics, Kyoto University,

Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, JAPAN

yanliang@db.soc.i.kyoto-u.ac.jp, {qiang,yoshikawa}@i.kyoto-u.ac.jp

Abstract Entity retrieval is a task to find information related to real-world-entities (person, organization. location. product, etc.). In this paper, we focus on the issue of spatio-temporal entity retrieval on Twitter, which searches for entities related to a given entity at a certain time in a certain region. For a given query, like “where can I buy iPhone4S now”, we capture the ERE (Entity-Relation-Entity) tuples from tweets and compute their scores to find the related entities (shops in our example) by considering: 1) terms which describes relations between entities or status and behaviors of entities, 2) properties of twitter users, 3) sentence structure, 4) descriptive polarities on entities and 5) time and location of the user who issued that query. We have designed the experiments for verifying the proposed method.

Keyword Entity Retrieval, Relation Mining, Spatio-Temporal Retrieval, Twitter

1. Introduction

Nowadays, with the rapid development of Internet, we can get more and more information from the Web. There are many websites which provide shopping information, such as kakaku.com. This kind of websites provides information on products and services for discussion, and comparison. They put shops’ information or make advertisements on the websites and then, users can use the provided information directly. However, it is still not easy to satisfy users enough, because the websites couldn’t update information so quickly and painstakingly. For example, from October of last year, iPhone4S appeared on the market and became so popular quickly that a lot of stores ran out of stock. The shopping website didn’t update the inventory information promptly enough to satisfy consumer’s purchasing needs. When a user asks a question of "where can I buy iPhone4S now in Kyoto", the answer user want is a list of shops in Kyoto where can buy the product iPhone4S “now”. “Now” implies a certain time and "in Kyoto" means the certain region, neither of them can users get directly from the websites we have introduced above and such a time-value issue cannot be solved easily by traditional keyword search.

Such task is an instance of Entity Retrieval. Entity Retrieval has become a new research area [1]. Unlike the traditional information retrieval for document, Entity Retrieval sets the focus level of retrieval one more step higher that it allows to search and rank named entities (person, organization, place, etc.) included in any kinds of text sources. What’s more, our query (e.g., where can I buy iPhone4S now in Kyoto) is not a simple entity retrieval query because it requests the entities with specific time and location. We define this task as ***spatio-temporal entity retrieval***, a task to search for entities at a certain time and in a certain region.

To search such spatio-temporal entity, for example, a list of shops, we need an information source which contains a mass of time-valueds messages. We choose Twitter, one of the popular micro-logging services; through which users talk about their daily experiences include a lot of shopping messages in real time, as the information source.

We propose an spatio-temporal entity retrieval method by taking the advantage of Twitter in this paper. For a given query like “Where can I buy iPhone4s now in Kyoto”, which means a issue of “iPhone4S buyable Shops”, we search for a list of result entities with certain time and certain region from Twitter. We try to capture ERE (*Entity-Relation-Entity*) tuples text from tweets and find a list of related entities by computing the score among them. The first E is the given entity got from input and the last E is the result entities we find. R is the status or behaviors words which describes the relationship between given entity and result entities. We also consider user property, sentence structure and descriptive polarities on entities to help searching for spatio-temporal Entity. In addition, the location of the user who issued the query and time when s/he issued the query are also considered well in our method. We try to find the answer as close as possible with the issued time and region.

This paper is organized as follows. In Section 2, we introduce the related work. In Section 3, we introduce the apatio-temporal retrieval method. We give a brief introduction of the experiment under preparing in Section 4 and conclude this paper in Section 5.

2. Related work

Recently, the research about entity search has become an emerging area, for example, on some forums such as the Text Retrieval Conference (TREC). The main research content of TREC Entity Track is searching for entities and their properties [2]. There are two major tasks. REF Related Entity Finding and ELC Entity List Completion. The goal of REF task is to retrieve a list of entities of a target type (such as person, organization, place and so on) required, which have to be related to a given entity according to a statement of the relation. For this task, M.Bron et al. propose a ranking method, by considering the occurrence of an entity among the top ranked documents for a given query as a vote for the existence of a relationship between this and the entity in the query [3].

Kaptein et al. [4] used Wikipedia as a pivot for finding entities on the Web which make the hard web entity ranking task easier. They tested and found that exploiting the structure of Wikipedia can improve entity ranking effectiveness. Sautos et al. [5] also addressed this issue to search a list of entities related to a given entity. They proposed that to tackle this problem as a voting process and considered the occurrence of an entity as a vote for the existence of a relationship between the target entity and the given entity. They provided a good idea to achieve entity retrieval task but they didn't focus on the entities at specific time and specific location which we are trying.

What's more, the task of searching for an object that has a specific time and location also be tried by other researchers. Real-time event detection by Twitter user, who is regarded as social sensors [6], is a task to detect the earthquake real-time with inferred time and inferred place by Tweets, which can be faster than news and guarantee a sufficient accuracy. However, the target location is quite different between this work and our research. Their goal is computing the region earthquake occurred by Tweets' location, while we intent to find the shops by extract the clue of shops' location from the content of Tweets, which is the difference with the research about detecting earthquake.

T.Sakaki etal. [7] also propose method to detect real-time events from Twitter. They extract reports of sighting on famous persons. The location information includes the record of location when the tweet published and the place names extracted from the content of the tweet. Our research also needs the support of location information and time stamp information from Twitter, but our goal is to find entities witch related to a given entity by a certain relationship, not to find the famous people's information like this paper.

The problem handled in this paper is spatio-temporal entity retrieval on Twitter, which try to meet the needs of special time and special space of a searching entity by taking the advantage of Twitter.

3. Spatio-Temporal Entity Retrieval on Twitter

3.1 Task Definition

We define the task of spatiao-temporal entity retrieval on Twitter in this sub-section. A spatio-temporal entity retrieval query should contain three parts as follow:

I. *(e) a given entity*
II. *(R) a description of the relationship between the given entities and the target entities*
III. *(E) the provision of the target entities (result entities), which should contain a certain time and a certain region as the property of a target entity*

Figure 1 shows an example. Considering a query of "Where can I buy iPhone4S now", in our definition, "iPhone4S" is the given entity, relationship is "buyable" and target entities will be a list of shops closed to the user in the current time. It means, the query is a tuple like "iPhone4S buyable Shop".

In addition, as the extra delimitation of spatio-temporal entity retrieval task, the time constraints and space constraints also need to be added into the input pattern. Time constraints (for example "now") and space constraints (for example "in Kyoto") are important for Spatio-Temporal Entity Retrieval task.

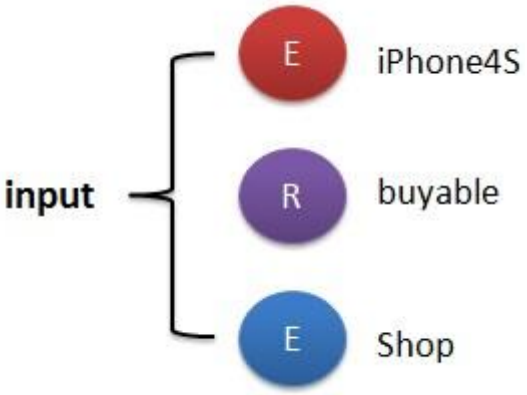

Figure 1: Query Structure

The output of this task is a ranked list of the target

entities. In other words, our goal is to find a list of target entities by using the three components in the input. As the example of "iPhone4S buyable shop", the output is a list of shops where the user can buy iphone4s now. The description of shops list should contain the shops' name and a visible address.

3.2 Basic Idea

The method contains five functional modules as Figure 3 shows: Search Twitter, Extends R set, Extract ERE tuple, Compute ERE score and Complete the address. We first search Twitter by using the input product as keyword, extend the R set, extract ERE tuples from the result tweets, find result entities by computing ERE score and output the completed information about result entities.

The main idea of the method to achieve this task is to extract the ERE (Entity-Relation-Entity) tuples from tweets and compute their relevance scores to the query. ERE tuple mean the given entity, the relation description and the target entity. Hereafter, without loss of generality, we explain our method by limiting the type of given entity and target one to product and shops, respectively. That is, we take it as an instance to handle while both product and shop are an instance type of entity.

To select the target entities, we compute the relevance score of each ERE tuple and the query by considering 1) terms which describes relations between entities or status and behaviors of entities, 2) user property, and 3) sentence structure, 4) descriptive polarities on entities, and 5) time and location of the user who issued that query. About this example showed by Figure 2, we need consider both the relation words and the distance among e-r-e.

3.3 Details of Our Method

(1) **Searching Twitter for Candidate Tweets:**

Firstly, we get the three phrase of a query: a product name, a relationship description, and a provision of E. We use the name of the product as a keyword to search the Twitter for a set of tweets which contains the name of the product and limit the search region according to the address of user. All of the result tweets will contain a created time.

Since iPhone4S appeared on the market from October last year, we found the period people keen on talking about inventory information is about two months from October, 2011 to November, 2011. Topsy Social Media Search Engine also allows us to set a time interval to limit the search results, which satisfy our needs for real-time search.

We collected all the tweets from October, 21th, 2011 to November, 20th, 2011 by "iPhone4S" as keyword. The period is just one month and our experiment will base on the data of that month.

The number of tweets which contains "iPhone4S" and published between October, 21th, 2011 and November, 20th, 2011 is 8409. The content of the tweets shows user's attention about iPhone4S. They talk everything about iPhone4S, such as speed measurement, the compare between KDDI Company and Softbank Company, even their opinions after using iPhon4S. Also, we do find some user writhed that they got iPhone4S from some shops, which is the valuable data in our experiment for the query "iPhone4S buyable shops".

Generally, the data we collected is not just the content of the tweets. We also got the user names and the published time of the tweets. The example of data showed as figure 3.

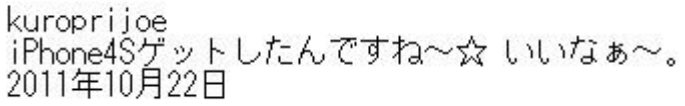

Figure 2: Data Pattern

As Figure 2 shows, our data include three elements; they are User Name, Content and the Date. Like the example showed in the figure, "kuroprijoe" as User Name, "iPhone4S ゲットしたんですね～☆ いいなぁ～。" as Content and "2011 年 10 月 22 日" as Date consist a date pattern.

(2) **Extenting R Set:**

R is the relationship between the given entity and the target entity, and R set is defined as a series of word which describing status and behaviors of the relationship as follow.

$$R(r_1, r_2, ..., r_k, ..., r_n) \quad (1)$$

For example, "can buy", as an instance of the R set, describe a status that buyable.

Generally, in a given query, R will be described by user with an instance of R set. Of course, there should be many words in the R set which can express the similar meaning or related to similar status, such as "can get", "on sale", the similar expression with "can buy", can be put into the same R set. They can be noun, adjective, verb and any other type of words or phrase.

We also use from WordNet or some other

language stools to get a list of synonyms of the given relationship description.

In addition, the structure of the sentence, the ERE computing model and the descriptive polarity consistency can also lead us to find the instances of R set.

We put three standards to judge if a word or phrase can be an instance of R set.

I. the instance of R set must be a word or a phrase which describe a status
II. In the sentence, the instance of R set must be in the similar place with given entity as the question
III. the instance of R set must have the similar descriptive polarities with the r from input

We will introduce the three standards in details.

Firstly, the instance of R set must be a word or a phrase which describe a status. We suppose a word which describes a kind of status would be a noun, an adjective or a verb in past tense. We use a language tool called “Juman” to analysis the Japanese words with showing the part of speech. We choose only noun, adjective or verb in past tense to be results.

Secondly, in the sentence, the instance of R set must be in the similar place with given entity as the question. We consider the Syntax structure of the sentence. Since the positional relationship between given entity and R is important, we analysis the structure of the sentence and extract the positional relationship between given entity and R. In order to improve efficiency, we just extract a short sentence near the given entity since we suppose the relationship word should near by the given entity. As Figure 3 shows, we extract the words near the given entity and analysis them to know their part of speech. Then, we choose the words which have the similar positional relationship with given entity as Figure 4 shows. The question in Japanese can be “iPhone4S を買える店はなんですか”, and the relationship word would be “買える” and it follows the given entity iPhone4S with a preposition “を”. So, we support the other relationship words would also follow the given entity with a preposition “を”, “は” or “が”, or without a preposition. As Figure 6 shows, “在庫ある” and ”ゲットした” have the similar positional relationship with “買える”.

Figure 3: Extract the Short Sentence

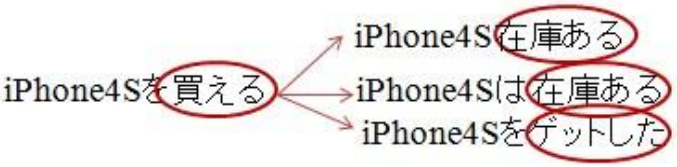

Figure 4: Positional Relationship Matching

Finally, the instance of R set must have the similar descriptive polarities with the r from input. Firstly, we need analysis the descriptive polarities of the input R. When the input R is “買える”, we suppose the word which express the meaning of “able” is a positive word that it shows a status that something is allowed to do. And the other instance of R set also need to be positive. As Figure 5 shows, “ゲットした” and “在庫ある” can be a positive word while “在庫なし” cannot be an instance of R.

Figure 5: Positional Relationship matching

(3) **Extracting ERE Tuples:**

Since we try to gather an R set first, the focal point when we extract ERE tuples is to find the target entity related to both given entity and the relationship describe words.

We firstly search in the data set by keyword “given entity” and “relationship describe words”. For example, we use “iPhone4S 在庫あり” as a search keyword, while “在庫あり” is an instance from R set.

Figure 6 shows an example of ERE tuple.

Figure 6: ERE Tuple

When we get a tweet, "池袋ビック Iphone4s 在庫あり". We can extract a ERE (given entity as "Iphone4", relational describe word as "在庫あり" and target entity as "池袋ビック") tuple. The relational describe word must be an instance of R set. Here is a simple introduction of the Mathematical calculations method of ERE score to extract the ERE tuple.

Different from the example, we suppose there maybe more than one ERE tuples in a tweet. We define the correlation degree of a ERE tuple as c_{e_1,r,e_2}, while e_1 must be the given entity and the type of e_2 must the same with the input. Since every ERE tuple will have a c, there may be more than one c from a tweet. We treat the c list from one tweet as follow.

$$C(c_1, c_1, ..., c_m) \quad (2)$$

We also suppose the distance between an entity and a relationship describe word as $d_{e,r}$, while e can be both given entity and target entity. The formula of C_{e_1,r,e_2} will be as follow.

$$C_{e_1,r,e_2} = 1/(d_{e_1,r} * d_{e_2,r}) \quad (3)$$

The $d_{e,r}$ will be decided by both the text distance and the punctuation between them. It means, the correlation degree of a ERE tuple decided by the distance between given entity and relationship describe word and the distance between target entity and describe word. The small both of the d is, the bigger C will be. We will select the ERE tuple with the biggest C from one tweet as an answer.

At the same time, we need consider the problem of classifying user to help us extract the ERE tuple. Twitter's user can be a person, a company, a team and so on. For our task to search a list of shops related to a kind of product, generally, the Twitter's user can be 1) an ordinary customer who is interested in the product and 2) official accounts of the shops or agents who sale the product.

We found that, the tweets published by the two kinds of users are really different from each other in whether the way to expression or using words. Specifically, the users of shops or agents more likely to publish advertisements from which we can easily to find the shops message such as address.

On the other hand, the tweets from ordinary customers are totally irregular. What's more, the tweets of an official user created may not contain the shops' message, but we can complete it easily from the user's profile.

Therefore, the method to process information of the two kinds of users can be quite different so we need to classify the users into two groups.

When users create their IDs, they can input the profile message include a self-description and an address, which give us a chance to identify their property and put them into a right group. Usually, shops' users will give an introduction of the shop which includes shop's name, address, telephone number and some other public information. We handle the classification problem by detecting the profile of the users.

Figure 7 and Figure 8 shows an example when we couldn't find a shop from the tweet. Figure 7 shows an example that cannot find a shop. Then, we check if the user is a seller for we can find a target entity from the profile of a seller user. Figure 7 shows the user's profile of the user in Figure 8. It is an example of a seller user that we find the target shop successfully from the seller user's profile.

Figure 7: Without Shop

Figure 8: User's Profile

(4) **Completing the Address Information of E**:

Users may indicate a relevant shop which satisfies the request, but the information of the shop is not enough to get to know where the shop is actually. For example, user mentioned that s/he bought an iPhone4S in Softbank Shop, but s/he didn't say which one it is. We need more information about the shop for it is really a result returned to the user. In this case, we need the basic information of the user who wrote the tweet, especially the activities scope of the user to help us guess the region the shop belongs.

If we couldn't find enough address information of the shop from the tweet, we consider the region of the shop is around the address of user.We can use the user profile, the past Tweets of the user or the following relationship to help us know where the user is. For example, the user said that s/he got iPhone4S in Softbank and we found the user lives in Sakyoku, Kyoto, then we can guess the most likely result list.

Figure 9 shows an example in the case that without enough location information. In the example, we have focused on "ビックカメラ" as a target entity, but we still don't know where the "ビックカメラ" is. We suppose that the "ビックカメラ" is around the user's home have a greatest possibility and we try to speculate where the user lives. We search from the user's profile and the old tweets as Figure 10 shows. As the example, we can guess the user lives in "名古屋市" and have a great possibility that he/she lives around "新栄 1 丁目". Finally, we use web map search engine such as Google Map and list the "ビックカメラ" shops that around "新栄 1 丁目".

Figure 9: An Example that Couldn't Find Enough Location Information

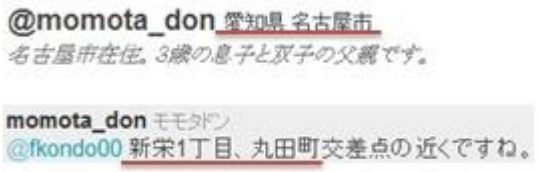

Figure 10: Location Information Found from User Profile and Old Tweets

4. Experimental Plan

The experiment is still in progress. We will introduce the part we have done and give a simple plan for the future experiment.

We use a social media search engine named Topsy [8] to collect tweets instead of Search Twitter Engine [9] and We have collected the data set between October, 21th, 2011 and November, 20th, 2011 by "iPhone4S" as keyword. We also selected some words and phrases to be candidates to consist an R set and we count the number of occurrences of each member of the R set. The future experiment procedures will be completed base on the method we introduced.

Also, we will verify the feasibility and efficiency of our method by the experiment. We plan to get a result just by person reading the tweets and evaluate the result by comparing the experimental results by the method and the artificial result.

5. Conclusion

This paper presents a spatio-temporal entity retrieval method on Twitter. Spatio-temporal entity retrieval is a special entity retrieval problem and choosing Twitter to be the data source and the search platform may make scene in this field. We extract the ERE (Entity-Relation-Entity) tuples from tweets and compute their relevance scores to the query to find a list of target entities, by considering terms which describes relations between entities or status and behaviors of entities, user property, sentence structure, and descriptive polarities on entities. In future work, we will complete the experiment to verify and improve the

proposed method.

6. Acknowledgement

This research is partly supported by the Scientific Research Grant(No.20300042 , No.20300036) made available by MEXT, Japan.

Reference

[1] M.Sayyadian, A.Shakery, A.Doan, and C.Zhai, “Toward Entity Retrieval over Structured and Text Data”, Proc. of ACM SIGIR 2004 Workshop on Information Retrieval and Databases, pp. 47—54, 2004.

[2] K. Balog, A. P. de Vries, P. Serdyukov, P. Thomas, and T. Westerveld. “Overview of the TREC 2009 Entity track”. Proc. of TREC 2009, 2009.

[3] M.Bron, K.Balog, and M.D.Rijke, “Ranking Related Entities: Components and Analyses”, Proc. of CIKM 2010, pp. 1079--1088, 2010.

[4] R.Kaptein , P.Serdyukov , A.D.Vries , and J. Kamps, “Entity ranking using Wikipedia as a pivot”, Proc. of the 19th ACM international conference on Information and knowledge management, pp. 69--78, 2010.

[5] R.L.T.Santos, C.Macdonald, I.Ounis, “Voting for Related Entities”, Proc. of RIAO '10 Adaptivity, Personalization and Fusion of Heterogeneous Information, pp. 1--8, 2010

[6] T.Sakaki, M.Okazaki, Y.Matsuo, “Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors”, Proc. of WWW2010, pp. 851--860, April 2010.

[7] T.Sakaki, Y.Matsuo, “Research on Real-time Event Detection from Social Media Prototype of Sighting Information Detection System”, Proc. Of The 25th Annual Conference of the Japanese Society for Articial Intelligence, 2011

[8] Topsy Social Media Search Engine, http://topsy.com/, 2012

[9] Search Twitter Engine, http://search.twitter.com, 2012