Context Based Semantic Relations in Tweets

Ozer Ozdikis, Pinar Senkul and Halit Oguztuzun

Abstract Twitter, a popular social networking platform, provides a medium for people to share information and opinions with their followers. In such a medium, a flash event finds an immediate response. However, one concept may be expressed in many different ways. Because of users' different writing conventions, acronym usages, language differences, and spelling mistakes, there may be variations in the content of postings even if they are about the same event. Analyzing semantic relationships and detecting these variations have several use cases, such as event detection, and making recommendations to users while they are posting tweets. In this work, we apply semantic relationship analysis methods based on term co-occurrences in tweets, and evaluate their effect on detection of daily events from Twitter. The results indicate higher accuracy in clustering, earlier event detection and more refined event clusters.

1 Introduction

Social networking platforms, especially micro-blogging sites such as Twitter, act as an important medium where people share their opinions with their followers. With its millions of users around the world and half a billion tweets per day,[1] textual content in Twitter is an abundant and still growing mine of data for information retrieval

[1] http://news.cnet.com/8301-1023_3-57541566-93/report-twitter-hits-half-a-billion-tweets-a-day

O. Ozdikis (✉) · P. Senkul · H. Oguztuzun
Middle East Technical University, Ankara, Turkey
e-mail: ozer.ozdikis@ceng.metu.edu.tr

P. Senkul
e-mail: pinar.senkul@ceng.metu.edu.tr

H. Oguztuzun
e-mail: oguztuzun@ceng.metu.edu.tr

F. Can et al. (eds.), *State of the Art Applications of Social Network Analysis*, Lecture Notes in Social Networks, DOI: 10.1007/978-3-319-05912-9_2,

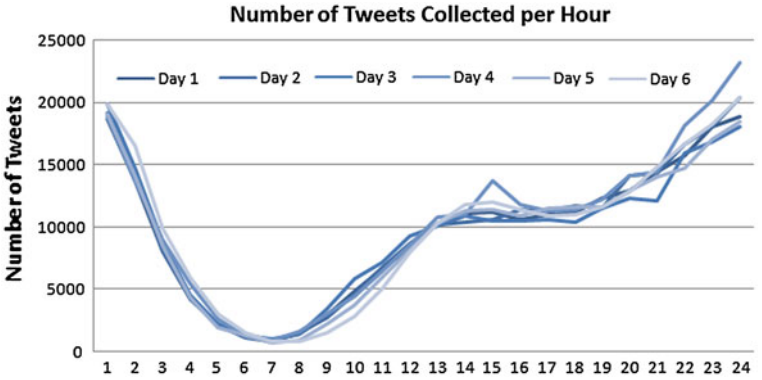

Fig. 1 Number of Tweets per hour collected by using the Twitter streaming API

researchers. This retrieved information is used for analyzing public trends, detecting important events, tracking public opinion or making on-target recommendations and advertisements. However, there may be variations in the contents of such a free and unsupervised platform, even if people refer to the same topic or the same concept in their tweets. Users may express the same thing in their own personal ways, such as by using unusual acronyms or symbols. Geographic, cultural and language diversities cause variations in the content. Even the regional setting or character set used on the device for posting affects uniformity. Moreover, the character limitation of a tweet in Twitter forces people to write in a compact form, possibly with abbreviations or symbols. Last but not least, spelling mistakes are definitely another major reason for divergence in content. For these reasons, in order to apply information retrieval algorithms more effectively in such an environment, it is necessary to be able to figure out the semantic relationships among the postings under the variations. Our goal is to identify such cases, and understand what the user could have meant, so that we can enrich a given tweet with possible similar terms.

In this work, we devise methods to extract semantic relationships among terms in tweets and use them to enhance the event detection capability. For the extraction of semantic associations, we use co-occurrence based statistical methods. Although the techniques we use are independent of language, we limit our scope to Turkish tweets posted in Turkey. In Fig. 1, we present a chart that displays the number of tweets per hour with Turkish content that we collect for several days. This figure indicates that, the collected tweets show a daily pattern and there are almost no postings around 7 in the morning. Therefore, we perform semantic relationship analysis and event detection on daily basis, such that we consider 7 am as the end of the day and identify hot topics of the previous day offline.

Our intuition is that implicit similarities among terms are time-dependent. In other words, consistently with the tweet graph in Fig. 1, we observe that a new day mostly starts with new events and new terms in Twitter. Therefore we choose to analyse term relations, i.e., co-occurrences, within the scope of per day. Using such a context

based relation extraction and applying these relations for tweet expansion, we aim to obtain earlier event detection, with longer lifetime and accurately clustered tweets. Moreover we obtain more refined results so that the users can follow the daily reports more easily. The rest of this chapter is organized as follows:

- We first present several metrics for measuring the associations among terms in a collection of documents (Sect. 2).
- We present our algorithm for event detection and introduce our proposed methods for enhanced event detection (Sect. 3).
- We provide analysis results by evaluating proposed techniques for discovering term associations and enhanced event detection. We discuss the results of our experiments (Sect. 4).
- We present a review of the related work about event detection, especially on social media. Recent studies using semantics in word co-occurrences and use of similarity analysis in different problem domains are given (Sect. 5).
- We finally conclude the chapter with an overview of the results obtained and future work directions (Sect. 6).

2 Term Relationship Metrics

In natural languages, terms can have several types of semantic and grammatical relationships in sentences. For example, a term can be the synonym, antonym, or hyponym of another term. Two closely related terms, such as the first and last names of a person, may occur in sentences more frequently than other term pairs. There are also phrases that are composed of multiple terms frequently used in the same pattern, such as "*take advantage of*", "*make use of*" or "*keep an eye on*". It is possible to look for these relationships in dictionaries, thesauri or encyclopedia. On the Internet, there are even online resources for this purpose such as WordNet and Wikipedia, which are already utilized in studies on similarity analysis and word sense disambiguation [2, 16]. However, in addition to the fact that these online resources are not mature enough for all languages yet, the language of Twitter is remarkably different than the one in dictionaries or newspaper texts. First of all, there is no authority to check the correctness of the content in Twitter. It is a social networking platform where people can write whatever they want in their own way. In its simplest terms, they can make spelling mistakes or they may type a word in different ways (like typing *u* for *ü* or *o* for *ö* in several languages). Therefore, instead of utilizing an online dictionary, we adopt a statistics-based technique to identify associations among a given set of terms. The idea is that semantic relations of terms have some impact on their distribution in a given document corpus. By analyzing syntactic properties of documents, i.e., tweets in this case, associations between term pairs can be extracted. There are several relationship metrics depending on which statistical patterns in term distributions to look for. First order relations, also known as syntagmatic relations, are used to identify term pairs that frequently co-occur with each other [19, 25].

A person's first and last names, or place names such as *United States* or *Los Angeles* can be considered to have this kind of relationship. Moreover, term pairs like *read-book*, *blue-sky*, and *happy-birthday* are examples of first order associations.

Term co-occurrences can be used in order to identify second order relationships as well. Second order associations are referred to as paradigmatic relations and they aim to extract term pairs that can be used interchangeably in documents [19]. If two terms co-occur with the same set of other terms, this can be interpreted as one can be replaced with the other (possibly changing the meaning of the sentence, but this is immaterial). Therefore, methods that find paradigmatic relations do not directly use co-occurrence counts between two terms, but consider the mutuality of co-occurrences with other words. For example, *photo-photograph* or *black-white* are such word pairs that most probably co-occur with the same words.

In addition to the first and second order associations, there are also higher order associations with basically the same logic [5]. For example, if there is a high number of co-occurrences among the term pairs $t_1 - t_2$, $t_2 - t_3$ and $t_3 - t_4$, then t_1 and t_4 can be considered as having a third-order association. In this work, we focus on first and second order relationships. Finding these relationships can be achieved by using several metrics. For syntagmatic relations, a straightforward measurement is simply counting the co-occurrences of term pairs. Other co-occurrence ranking metrics are proposed such as Dice, Cosine, Tanimoto and Mutual Information [10]. Application of entropy-based transformations [20] or Singular Value Decomposition [7] on the co-occurrence frequencies are also further improvements for first-order relations. For finding second order relationships between two terms t_1 and t_2, one of the metrics is to count the number of distinct terms t_3 that co-occur with t_1 and t_2 [5]. However, in our work, we apply a method based on the comparison of co-occurrence vectors as presented in [19]. The basic idea is to generate term co-occurrence vectors and compare their similarity in the vector space. We experiment with cosine and city-block distances for similarity comparisons. For first order relations, we simply count the number of co-occurrences, i.e., raw co-occurrence values. Both in the first and second order association analysis, our objective is not only finding the most related terms, but also assigning them a similarity score, i.e., a value between 0 and 1. This similarity score will be used while applying a lexico-semantic expansion to tweet vectors, as will be explained shortly.

Statistical methods do not necessarily require a dictionary or human annotated data. First of all, this brings about language independence. Moreover, by analyzing terms in specific timeframes, ambiguities can be resolved depending on the context. For example, the term *goal* could have many related terms, such as *match*, *objective* or *end*. But if we know that there was an important soccer game that day and the term appears very frequently in tweets, then we can assume that it is used in sports context, therefore *match* would be the most suitable related term.

Regarding the performance issues and resource limitations, we do not apply semantic analysis on rare terms. On a daily collection of around 225 K tweets, there are more than 140 K distinct words on average, some of them appearing only in a few tweets. Moreover, in order to capture co-occurrences, content-rich tweets are preferred. Therefore, we process tweets with at least 4 terms and compare the terms with

minimum frequency of 300. These numbers are selected by intuition, after observing the Twitter traffic for a period of time. They can be adapted for another language, another tweet selection criterion, or another processing infrastructure. Here we would like to emphasize that our focus in this work is to demonstrate that the extraction of semantic relations in an uncontrolled environment such as Twitter is practical for better event detection. Finding the most suitable parameter values or most efficient similarity metrics could be the objective of another study.

2.1 First Order Relationships

As explained before, in order to find the first order relationships, we use the raw co-occurrence values. In our previous work [13], after finding the number of times the term pairs co-occur, we identified the semantically related term pairs if they appear in more than 50 tweets together. Moreover, if two terms are found to be semantically related, we assigned a constant similarity score of 0.5. In this work, we developed a more generic solution and adopted the approach that we used for discovering hashtag similarities in [14]. Instead of using a threshold for deciding the similarity of two terms and giving them a constant similarity score, we assign normalized similarity scores for each term pair by using their co-occurrence values. For example, the term pair with the maximum number of co-occurrence value, c_{max}, on a given day has the similarity score 1.0. For other term pairs t_i and t_j with a co-occurrence count of $c_{i,j}$, their similarity score is given by the ratio of $c_{i,j}/c_{max}$.

2.2 Second Order Relationships with Cosine Similarity

For the second order relationships, term co-occurrence vectors are generated. Let $c_{i,j}$ represent the number of co-occurrences of the terms t_i and t_j. Then, for each term t_i, we count its co-occurrences with other terms t_1, t_2,... $t_{|w|}$ where W is the set of distinct terms collected on that day's tweets. After forming the term vectors as given in (1),

$$\mathbf{t_i} = (c_{i,1}, c_{i,2}, ..., c_{i,i-1}, 0, c_{i,i+1}, ..., c_{i,|W|-1}, c_{i,|W|}) \tag{1}$$

we compare their cosine distance by using the cosine distance equation in (2) [28].

$$sim_{\text{cosine}}(\mathbf{t_i}, \mathbf{t_j}) = \frac{\mathbf{t_i} \cdot \mathbf{t_j}}{|\mathbf{t_i}||\mathbf{t_j}|} = \frac{\sum_{k=1}^{|W|} c_{i,k} c_{j,k}}{\sqrt{\sum_{k=1}^{|W|} c_{i,k}^2 \sum_{k=1}^{|W|} c_{j,k}^2}} \tag{2}$$

Again we do not use any threshold for the decision of similarity but rather use the cosine distance as the similarity score, which is already in the range [0,1].

2.3 Second Order Relationships with City-Block Distance

City-block distance is another simple vector comparison metric [19]. After forming the co-occurrence vectors, while comparing two vectors, it finds the sum of absolute differences for each dimension as given in (3).

$$sim_{city\text{-}block}(\mathbf{t_i}, \mathbf{t_j}) = \sum_{k=1}^{|W|} |c_{i,k} - c_{j,k}| \tag{3}$$

Similar to the solution we applied for first order relations, we normalize the distances in [0, 1] and use these values as similarity scores.

3 Event Detection and Semantic Expansion

In this work, we perform offline event detection on tweets. However the algorithms we implement can also be used online with further performance optimizations. The flow of the event detection process is depicted in Fig. 2. Dashed arrows indicate the extension that we implemented on a traditional clustering algorithm. We first present the data collection, tweet vector generation, clustering and event detection steps. Then we explain how we carry out lexico-semantic expansion and improve event detection quality.

For tweet collection from the Twitter Streaming API, we use Twitter4J,[2] a Java library that facilitates the usage of Twitter API. We apply a location filter and gather tweets posted by users in Turkey, with Turkish characters. Posts with other character sets such as Greek or Arabic letters are filtered out. The gathered tweets are immediately stemmed with a Turkish morphological analyzer called TRMorph [6]. After further preprocessing, including the removal of stop words and URLs, they are stored into the database. Using this process, we collect around 225 K tweets per day. Further details regarding the tweet collection and preprocessing steps are found in our previous work [13].

Our event detection method is an implementation of agglomerative clustering, applied on tweets collected in a given period of time. In this algorithm, tweets are represented by tweet vectors that are generated by the TF-IDF values of the terms in each tweet. In order to fix the size of these vectors and calculate the IDF values of terms, all tweets in the given period are pre-processed. The number of distinct terms, i.e., the dimension of tweet vectors, is determined and document frequencies of each term are found. Finally tweet vectors are created by using the frequencies of their terms and their inverse document frequencies.

The clustering algorithm simply groups similar tweets according to the distance of their tweet vectors in n-dimensional vector space. Just like tweet vectors, clusters

[2] Twitter4J homepage, http://twitter4j.org.

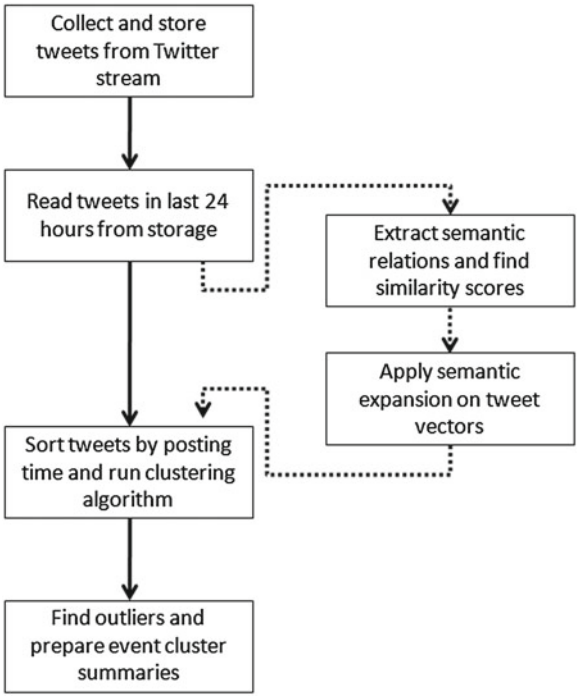

Fig. 2 Event detection process

are also represented with vectors. A cluster vector is the arithmetic mean of the tweet vectors grouped in that cluster. Tweets are processed one by one according to their posting time. For each tweet, the most similar cluster vector is found. For similarity calculation, we use cosine distance as given in Eq. (2). If the similarity of the most similar cluster is above a given threshold, the tweet is added to that cluster and the cluster vector is updated. Otherwise, that tweet starts a new cluster on its own. If no tweet gets added to a cluster for a certain period of time, then it is finalized, meaning that the event does no longer continue.

Finally, with all tweets processed, we apply an outlier analysis using the empirical rule (also known as the three-sigma or 68-95-99.7 rule) [14, 29]. According to this rule, we first find the mean number of tweets in clusters and their standard deviation (σ). We mark the clusters with more than mean+3σ tweets as *event clusters*. In order to present an understandable summary to the user, summaries are generated for each event cluster. Summaries are simply the first three terms in cluster vectors with the highest TF-IDF values.

The event detection method we introduced above is our basis method, whose path is indicated with the solid arrows in Fig. 2. We refer to it as BA (Basis Algorithm) in the rest of the chapter. The novelty of our work is an extension on this basis algorithm by applying a lexico-semantic expansion to tweets. This extension is composed of two

steps, namely calculation of similarity scores among frequent terms and using these scores while applying the semantic expansion on tweet vectors before feeding them to the clustering process. As explained before, we implemented three metrics for the analysis of term associations. We evaluate their results and apply them on clustering separately. We label these clustering implementations as FO for First Order, COS for Cosine and CBL for City-Block metrics.

The first step in this extension is the calculation of similarity scores among term pairs that appear in the tweets to be clustered. By using one of the abovementioned metrics, we obtain term–term similarity scores. For each term, we keep only top-n similarities, due to the performance optimizations. We choose to use top-3 similarities in our experiments.

After having the similarity scores, generated either with first order or second order analysis, we use these similarity scores to apply semantic expansion on tweet vectors. The idea of semantic expansion is similar to the studies in [9] and [16]. In this work, we develop specialized methods for evaluating the numerical values. The expansion process is as follows:

1. Given a tweet t^i and its tweet vector with k terms $[t^i_1, t^i_2, ...t^i_k]$ with corresponding TF-IDF weights $[w^i_1, w^i_2, ...w^i_k]$,
2. For each t^i_x in the tweet vector,
    a. Search for its semantically related terms. Let t^i_x be associated with the three terms t_a, t_b, t_c with similarity scores s_a, s_b, and s_c
    b. Find the products $w^i_x s_a$, $w^i_x s_b$, and $w^i_x s_c$ as expanded TF-IDF values
    c. If t_a does not exist in the tweet vector or if its TF-IDF value in the tweet vector is less than the expanded TF-IDF value, namely $w^i_x s_a$, then insert t_a to the tweet vector with its expanded TF-IDF value. Otherwise, simply ignore it. That means if a term already exists in the tweet with a high TF-IDF value, then it is not changed by the semantic expansion process. Do this step for t_b and t_c as well.

Such an expansion usually results in tweet vectors with much higher dimensions than the original ones. The effect of this may be larger clusters with more similar tweets, or larger clusters with junk tweets. Therefore it is important to identify correct term similarities with correct similarity scores. As elaborated in the following section, application of such a semantic expansion has several advantages.

4 Evaluation and Results

Our evaluations focus on two core functions implemented in our enhanced event detection method, namely the success of identified term similarities and the quality of detected events. Our 3-week test dataset is composed of about five million tweets posted between the 4th and 25th of September 2012 with Turkish content in

Turkey. We adopt daily processing of tweets, i.e., tweets collected during one day are processed at 7am in the following morning. The results are evaluated in accordance with this regular daily processing, presented for each day covered by our test dataset.

4.1 Evaluation of Term Similarity Analysis

Before presenting the similarity analysis, we present some figures to clarify the statistical value of our evaluation. The average number of tweets per day, collected during the generation of our test dataset is 225,060. On average, 140 K distinct terms are used in Twitter every day, with 518 of them deemed as having high frequency by our definition (>300). It means that we compare about 518 terms on average every day for their semantic associations.

For evaluating the success of a semantic relationship extraction method, it is possible to make use of word similarity questions in a language exam such as TOEFL [19, 20]. Comparing the results of the algorithm with the answer sheet of the exam is an acceptable method. However, terms written in Twitter are hardly found in a dictionary due to the variances in writing conventions and spelling mistakes. Moreover, ambiguous similarities can exist depending on the context, as we have explained in an example where the term *goal* must be related to the term *match* if there is a soccer game that day. Discovery of such context-based associations may not always be captured by an automatic language test.

In order to set a golden standard for evaluation, we prepared a questionnaire per day that is composed of 120 questions for each day (2,520 questions in total). They were five-choice questions, where for a given term, users were asked to mark the most relevant term among the choices. Choices for each question are populated from the results of three similarity measurement algorithms (the ones with highest similarity scores) and two randomly selected terms from the corpus. If similarity estimations of two algorithms coincide, we write that term only once in the choices, and another random term is inserted to present five distinct choices for the question. Questions are answered by seven users who are native Turkish speakers and mostly computer science graduates. We explained them briefly our relevance criteria, such as spelling mistakes (*günaydın* $\sim$ *gunaydin*), abbreviations (*Barca* $\sim$ *Barcelona*), and strong association of domains (*sleep* $\sim$ *dream*). Users could also select "none", if they think there is no related term among the choices or if the term in the question may not have a matching term (e.g. it can be a number, a symbol, or some meaningless word which makes it impossible to decide similarity). Therefore, unanswered questions could mean that either it is inapplicable, or none of the similarity algorithms could successfully find a similar term. We compare the similarity algorithms only on the answered questions.

While grading the success rates of the algorithms, we count the number of correctly estimated similar terms. If the term with the highest similarity score found by an algorithm is the same as the term marked by the user, it is accepted as a correctly answered question by that algorithm. There may be cases where all three algorithms

Table 1 Offline term similarity results

Day	Marked questions	Accuracy ratio		
		FO (%)	COS (%)	CBL (%)
1	63	**62**	25	25
2	53	**38**	36	30
3	73	**44**	36	34
4	66	35	**50**	15
5	68	**44**	28	21
6	68	**49**	32	25
7	73	**52**	27	29
8	84	**63**	19	29
9	77	**38**	36	27
10	62	**61**	31	21
11	58	**48**	41	21
12	67	**49**	30	6
13	71	**56**	34	14
14	69	**54**	33	23
15	70	**54**	29	24
16	90	**47**	33	17
17	82	**41**	30	22
18	101	**42**	30	17
19	97	**48**	31	16
20	65	**43**	32	18
21	58	**48**	31	16
Avg.	72	**48**	32	21

find the same correct result for a question. Then they all get their points for that question. As a result, the ratio of the number of correct results of each algorithm to the number of marked questions is used to evaluate the accuracy of the algorithms. The results of term similarity evaluations are presented in Table 1. The column labeled with "Marked questions" indicates the number of questions with a valid answer, i.e., the user did manage to select an answer among the choices. The percentages of correct results for each algorithm are presented in the rest of the columns, where the highest accuracy ratio for each question is highlighted. Results are presented for each day in the test data set.

The number of answered questions shows that, users could find a relevant term among the choices for more than half of the questions. Ignoring the luckily selected random terms while generating the choices of the questions, this can be interpreted as at least one algorithm finds a relevant term in the corpus at least half of the time. According to this table, the first order relations are apparently closer to the users' answers. Several examples of successfully detected first order relations are *birth* $\sim$ *day*, *listen* $\sim$ *music*, and *read* $\sim$ *book*. Obviously, given the first term, these can be considered as the first mappings that come to mind even without any multiple

choices. In other words, they are easier to identify. Second order relations are a little harder for a person to see at first. Several examples of the correctly selected second order term pairs are *morning* ∼ *early*, *happy* ∼ *fine*, and *class* ∼ *school*. Therefore we believe the accuracy ratio of the second order associations should not be underestimated. Between cosine distance and city-block distance, cosine similarity makes more accurate guesses.

Although this table can give an idea about the power of co-occurrence based similarity techniques, calculated similarity scores play an important role for the event detection and expansion algorithm. The effect of similarity scores is better observed in the evaluation of event detection.

4.2 Evaluation of Event Detection

In the Topic Detection and Tracking (TDT) domain, evaluation of event detection algorithms is usually made either by precision-recall [23] or by false alarm-miss rate analysis [3, 8]. In our work, we use the precision-recall metric for evaluation. Moreover, we analyzed the event detection times and studied the refinement of the generated event clusters. The comparison of the event detection algorithms mainly focuses on the following three goals:

1. Number of event clusters and their tweets: In order to present a useful and readable report to the users, it is preferable to generate more refined event clusters with as many correctly clustered tweets as possible. The best case would be presenting unique event clusters for each actual event in the real world, with explanatory cluster summaries.
2. Accuracy of event clusters: Ideally, there should be no irrelevant tweet in an event cluster, and an event cluster should cover as many relevant tweets as possible. Therefore, our goal is to improve the accuracy of tweets in event clusters.
3. Time span of detected events: Our expectation is that the utilization of hidden semantic relations among terms should result in earlier generation of event clusters with longer duration. Especially longer duration of larger clusters shift their finalization to a later time. Therefore, they can attract unique tweets that would normally start new clusters on their own.

Our event detection evaluation is based on human annotated tweets. For each day in our test data set, we run four event detection algorithms, where our baseline is the one with no semantic expansion (BA). The results are compared with the other algorithms using different expansion methods, namely first order (FO), second order with cosine (COS) and city-block distances (CBL). An event detection algorithm may find several event clusters about an event. We consider them all as belonging to the same event, which means that an event may be represented by one or more event clusters in an algorithm. While matching an event cluster with an event, we consider its cluster summary. We would like to remind that the cluster summaries are in fact

the top three terms in the event cluster with the highest TF-IDF values. In order to assign an event cluster to an event, its summary must be understandable and clearly mention the corresponding event.

After the event clusters are obtained, we select one event per day as our "target event" for that day. While determining a target event cluster, we utilized other resources in Internet such as TV rating results or newspapers in order to determine the most popular and important event for each day. On the other hand, we observed that people are posting tweets that are not newsworthy, such as "good morning" or "good night" tweets. It is possible to apply a classification algorithm in order to filter out such event clusters [4, 22]. We simply let them survive as event clusters but do not consider them as target events in our evaluations.

Although our event detection algorithm is executed on a daily basis at 7 am, events do not have to last one day. According to our observations, almost no event lasts longer than two days though. In fact, 2-day events are observed only when the event happens at midnight. Since we process tweets at 7am, an event that happened around midnight is usually detected on both days. For example, the first target event in our test data set is the tragic loss of a young soccer player, Ediz Bahtiyaroglu, who had played in several major Turkish soccer clubs and passed away at the age of 26 from heart attack. As soon as this event is heard at night, people posted tweets to express their sorrow and condolences. These tweets continued during the day, which resulted in detection of this event the next day as well. Apart from this, other target events that we marked for evaluation can be classified as soccer games, anniversaries of historical events, disasters, criminal incidents, popular TV shows or news about celebrities, which are mentioned by thousands of people in Turkey.

We first present the number of event clusters, number of annotated tweets, and the results of our precision-recall analysis in Table 2 for each day in our test data. The table can be interpreted as follows. The first column labeled as "Day" displays the index of the day between 4th and 25th of September 2012 (e.g. Day-1 is the 4th of September). The row "avg" is the average of all days for the feature in the corresponding column. The second column, namely "Annt.", represents the number of tweets that are manually annotated by human annotators as belonging to the target event of that day. By "annotation", we mean manually marking a tweet to be related to an event or not. We adopt the following method to generate a tweet set to annotate. Given a day, each algorithm that we implement generates their event clusters for the target event for that day. Assume the set of tweets grouped by each of these algorithms are T_{BA}, T_{FO}, T_{COS}, and T_{CBL}. Then the tweets that we manually annotate are the union of these tweet sets, i.e., $T_{BA} \cup T_{FO} \cup T_{COS} \cup T_{CBL}$. Consider Day-20 for example. The target event of that day was the game of a popular sports club in Turkey. Each algorithm detected multiple event clusters for that target event, and the number of tweets clustered in these event clusters were 574, 427, 674, and 519 respectively for each algorithm. As shown in the table, the union of these tweet sets is composed of 862 tweets, which gives the tweets to be annotated for that day's target event.

The third column group that is labeled as "Target Event Cluster Ratio" displays the number of target event clusters and the number of event clusters detected for that

Table 2 Offline event detection results

Day	Annt.	Target event cluster ratio				Precision				Recall				F-Score			
		BA	FO	COS	CBL	BA	FO	COS	CBL	BA	FO	COS	CBL	BA	FO	COS	CBL
1	495	2/15	2/13	2/11	2/11	**1.00**	0.99	0.98	0.98	0.50	0.70	0.72	**0.78**	0.66	0.82	0.83	**0.87**
2	2356	2/12	2/4	2/9	3/7	**0.98**	0.48	0.93	0.94	0.30	**0.77**	0.68	**0.77**	0.46	0.59	0.79	**0.85**
3	3907	7/14	7/13	6/13	5/8	**0.97**	0.94	0.88	0.85	0.50	0.63	**0.79**	0.61	0.66	0.76	**0.83**	0.71
4	1314	4/21	5/17	2/16	2/14	0.99	0.88	**1.00**	0.91	0.45	**0.79**	0.26	0.32	0.62	**0.83**	0.41	0.48
5	561	1/18	1/7	1/11	1/12	**1.00**	0.66	0.74	0.79	0.24	0.71	0.71	**0.72**	0.39	0.69	0.72	**0.75**
6	2367	1/6	2/5	2/6	3/9	**1.00**	0.68	0.91	0.88	0.57	**0.92**	0.81	0.78	0.72	0.78	**0.86**	0.83
7	769	2/11	2/12	1/11	1/8	**0.94**	0.92	0.92	0.83	0.42	0.43	0.72	**0.88**	0.58	0.58	0.81	**0.86**
8	4014	7/15	7/12	8/12	11/16	0.59	0.54	**0.62**	0.59	0.51	0.58	**0.72**	0.66	0.55	0.56	**0.67**	0.62
9	288	1/16	1/13	1/15	2/16	**1.00**	0.95	**1.00**	0.78	0.57	0.62	0.57	**0.75**	0.73	**0.76**	0.72	**0.76**
10	800	1/14	1/14	2/11	1/10	**1.00**	**1.00**	0.92	0.92	0.53	0.51	0.49	**0.90**	0.69	0.67	0.64	**0.91**
11	869	2/14	0/4	3/17	3/15	0.64	NaN	0.65	**0.67**	0.38	NaN	0.74	**0.94**	0.48	NaN	0.69	**0.78**
12	1103	1/16	2/11	1/6	2/5	**1.00**	0.76	0.65	0.61	0.33	0.53	0.73	**0.81**	0.50	0.63	0.59	**0.70**
13	1404	2/9	2/8	2/9	5/12	**1.00**	**1.00**	0.97	0.98	0.66	0.66	**0.70**	0.65	0.80	0.80	**0.82**	0.78
14	822	2/10	1/6	1/11	2/6	0.46	**0.98**	**0.98**	0.44	0.45	0.63	0.60	**0.85**	0.46	**0.76**	0.74	0.58
15	1233	4/16	1/9	1/10	2/9	0.98	**1.00**	0.99	0.97	0.36	0.15	0.41	**0.72**	0.53	0.26	0.58	**0.83**
16	4550	5/11	2/4	5/9	5/8	**0.99**	0.84	0.96	0.90	0.47	0.58	**0.63**	0.51	0.64	0.69	**0.76**	0.65
17	4110	5/10	5/7	4/5	5/8	**0.99**	0.94	0.86	0.89	0.63	**0.76**	0.63	0.57	0.77	**0.84**	0.73	0.69
18	313	2/19	1/11	1/12	0/10	**1.00**	**1.00**	**1.00**	NaN	**0.91**	0.48	0.35	NaN	**0.95**	0.65	0.52	NaN
19	326	1/16	1/13	2/12	2/11	**0.99**	0.61	0.71	0.77	0.63	0.67	**0.93**	0.88	0.77	0.64	**0.81**	0.82
20	862	4/15	2/9	3/12	5/13	**1.00**	0.87	0.79	**1.00**	**0.81**	0.52	0.75	0.73	**0.89**	0.65	0.77	0.84
21	251	2/10	0/9	1/12	0/5	0.40	NaN	**0.99**	NaN	**0.92**	NaN	0.64	NaN	0.56	NaN	**0.77**	NaN
avg	1557	2.8/13.7	2.2/9.6	2.4/11.0	2.9/10	**0.90**	0.84	0.88	0.83	0.53	0.61	0.65	**0.73**	0.64	0.68	0.72	**0.75**

day. For example, on Day-20, 15 event clusters were identified as outliers by the BA algorithm. Among these 15 event clusters, we found four of them to be related with the target event, namely the soccer game. These numbers give an idea about the information contained in event clusters if considered together with the coverage. The lowest number of event clusters with the highest coverage ratio leads to more understandable event cluster summaries for people. Otherwise, if too many event clusters are generated with low coverage, this would mean scattered information with poor comprehensibility.

Rest of the columns in the table present the precision, recall, and F-score values for each day for all clustering algorithms that we implement. If an algorithm finds no event cluster for a target event, then its precision-recall analysis becomes inapplicable, denoted as NaN. This accuracy analysis can be interpreted as follows: The basis algorithm usually results in better precision, 0.90 on average. This is because we apply no semantic expansion to the tweets, and there is less chance for an irrelevant tweet to be included in a cluster. On the other hand, its coverage is not as large as the algorithms using semantic expansion. Whether it is the first or second order relationship, semantic expansion techniques usually cover more tweets than the basis algorithm. The overall accuracy is calculated by using the F-score equation given in (4). According to these results, second order associations provide higher accuracies.

$$Fscore = \frac{2 \times precision \times recall}{precision + recall} \tag{4}$$

In addition to these analyses, another criterion for better event detection is the time of detection. It is preferable to hear about an event as soon as possible. As an example, we present the active times of event clusters of Day-20 in Fig. 3. An active event cluster is depicted in tiles, with its beginning and ending times corresponding to the earliest and latest tweet times, respectively. The time window covered in each algorithm is highlighted in gray. The first group of rows is the target event clusters found by the BA algorithm. As given in Fig. 3, the BA algorithm detected four event clusters for the soccer game that day. Therefore the figure displays four lines for BA with the clusters' active time windows. According to this algorithm, the first tweet about the game was posted at around 14:00. In other words, the track of this event was first observed in the afternoon and lasted for about 2 h. Then, no similar tweet has been observed until the evening. It can also be said, the content of tweets after 16:00 were not very related to the first event cluster. Then at about 20:30, the first goal was scored in the game, which cause the generation of two event clusters at that time. The name of the scoring player was *Burak Yılmaz*, which was also spelled as *Burak Yilmaz* in some tweets (the letter *i* replacing *ı*). We consider this as the most important reason for two event clusters about the same event. The last event cluster generated by BA begins at 10 pm and lasts for three hours. The summary of this event cluster is about the celebrations after the end of the match.

In the first three algorithms, the event detection times are almost the same. On the other hand, the duration of the first event cluster is longer for FO and COS algorithms. This is considered to be the result of successfully identified semantic associations.

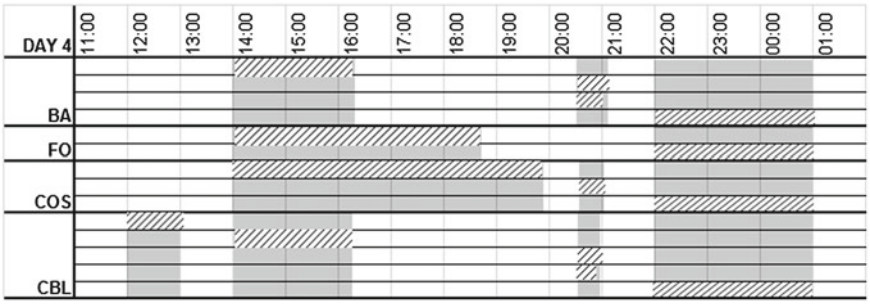

Fig. 3 Timespan of event clusters on day-20

Interestingly in the COS algorithm, there is only one event cluster detected at 20:30. Apparently the algorithm found a high similarity score between the terms *Yılmaz* and *Yilmaz*, which results in a single event cluster at that time. This observation highlights one of the major objectives of our research, i.e., elimination of duplicate event clusters. Another objective was earlier detection of events. In this specific case on Day-20, this is achieved by using the CBL algorithm. The first event cluster generated for this event starts at about 12:00, which is 2 h earlier than the earliest event detection times of other algorithms.

5 Related Work

Event Detection, also known as Event Detection and Tracking, aims to identify unique events by processing textual materials such as newspapers, blogs, and recently, the social media [1]. Especially after the foundation of Twitter in 2006 and with its millions of users around the world today, there have been many studies that utilize peoples' postings for information retrieval purposes [11, 12]. An implementation of real-time event detection from Twitter is described in [22]. In that work, Sankaranarayanan and coworkers follow tweets of handpicked users from different parts of the world; cluster them for event detection, and assign a geographic location to the event in order to display it on a map. In a similar study, Sakaki and coworkers focus on earthquakes in Japan [21]. They detect earthquakes and make estimations about their locations only by following tweets in Twitter. There are also studies for improving first story detection algorithms on Twitter, i.e., identifying the first tweet of a new event [17]. An interesting use of event detection in Twitter is presented in [15]. In that work, Park and coworkers aim to detect important events related to a baseball game, and display annotated information to people that watch that game on TV. This can be considered to be a similar case of our example in Fig. 3. By detecting separate events for scoring a goal, or making a homerun as stated in [15], it is possible to retrieve who made the homerun at what time and where.

Using the semantics in word co-occurrences has been exploited in several studies on event detection. In [23], authors implement a solution by integrating burst detection and co-occurrence methods. In that solution, they track a list of terms (entities which may be taken from a reference work like Wikipedia) in query logs or tweet data in order to detect extraordinary increases in their appearances. They argue that if two entities show unusually high frequencies (bursts) in the same time-window, they possibly belong to the same event. In order to measure their relevance and group in the same event, content-rich news articles in the corresponding time window are processed and first order associations among terms are analyzed. A similar approach is used in [24]. The intuition is that documents about an event should contain similar terms. By generating a graph with terms as nodes, and co-occurrences as edges, they identify highly connected sub-graphs as event clusters.

Apart from event detection purposes, similarity analysis on textual materials can be useful for recommendation and applying intelligent expansion to queries. In [27], authors study the spatio-temporal components of tweets and identify associations among trending topics provided by Twitter API. They generate vectors considering their spatial and temporal aspects, which are then pairwise compared with Euclidean distance to find the most similar topic pairs in Twitter. In a more recent work, methods in [27] are extended with the similarities of topic burst patterns in order to take event-based relationships into account [26]. The intuition is that two similar topics should have similar bursting periods as well as spatial and temporal frequency similarities.

A method for exploring associations among hashtags in Twitter is presented in [18]. The proposed model aims to make more complex temporal search requests in Twitter, such as asking for hashtags with an increasing co-occurrence with a given hashtag in a desired period of time. Another example of analyzing term associations is given in [10], where users are guided while giving a title for the product they want to sell in an online store. Text entered by a seller is compared with previous queries of buyers, and a better title for the product is recommended to the seller.

In our preliminary work, we presented the basics of the techniques that we elaborate in this chapter, executed on a very limited data set of three days [13, 14]. These techniques have been combined and extended for context-based daily event detection, tested on a much larger data set annotated by several users. Moreover, detailed evaluations focus on both term similarity analysis and event detection methods, providing a more reliable and clear overview of results.

6 Conclusion

In this work, we aim to extract associations among terms in Twitter by using their co-occurrences and use them in a semantic expansion process on tweets in order to detect events with higher accuracy, with larger time span, and in a user-friendly form. We improve our previous work by using similarity scores instead of thresholds and constant multipliers for semantic expansion. Moreover, we identify context-dependent associations by evaluating terms in specific time windows. Daily event

clusters are determined by making an outlier analysis. Although our methods are applied on tweets in each day, they can be adapted to work in different time granularities or in an online system, which we are planning to implement as a future work. Moreover, we would like to experiment periodically merging and/or dividing event clusters in the course of event detection in order to improve the resulting event clusters.

Our methods are tested on a set of around five million tweets collected in three weeks with Turkish content. We implemented three different semantic similarity metrics and evaluated them on this test data set. Results of these metrics are further analyzed in the evaluations of our event detection methods. Improvements are observed in event detection in several aspects, especially when second order associations are used. As the methods we implement do not require a dictionary or thesaurus, they can be used for other languages as well.

Acknowledgments This work is supported by TUBITAK with grant number 112E275.

References

1. Allan J, Carbonell J, Doddington G, Yamron J, Yang Y (1998) Topic detection and tracking pilot study: final report. In: Proceedings of the DARPA Broadcast News transcription and understanding, Workshop
2. Banerjee S, Pedersen T (2002) An adapted Lesk algorithm for word sense disambiguation using wordNet. In: Lecture notes in computer science, vol 2276, pp 136–145
3. Can F, Kocberber S, Baglioglu O, Kardas S, Ocalan HC, Uyar E (2010) New event detection and topic tracking in Turkish. J American Soc Inf Sci Technol 61(4):802–819
4. Castillo C, Mendoza M, Poblete B (2011) Information credibility on Twitter, In: Proceedings of World Wide Web conference
5. Chakraborti S, Wiratunga N, Lothian R, Watt S (2007) Acquiring word similarities with higher order association mining. In: Lecture notes in computer science, vol 4626
6. Coltekin C (2010) A freely available morphological analyzer for Turkish, In: Proceedings of the 7th international conference on language resources and evaluation (LREC)
7. Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990) Indexing by latent semantic analysis. J American Soc Inf Sci 41(6):391–407
8. Fiscus JG, Doddington GR (2002) Topic detection and tracking evaluation overview. In: Allan J (ed) Topic detection and tracking: event-based information organization. Kluwer Academic Publishers, Dordrecht, pp 17–31
9. Frasincar F, IJntema W, Goossen F, Hogenboom F (2011) A semantic approach for news recommendation, In: Business intelligence applications and the web: models, systems and technologies. IGI Global, Hershey
10. Huang S, Wu X, Bolivar A (2008) The effect of title term suggestion on e-commerce sites. In: Proceedings of the 10th ACM workshop on Web information and data management (WIDM), pp 31–38
11. Java A, Song X, Finin T, Tseng B (2007) Why we Twitter: understanding microblogging usage and communities. In: Proceedings of SNA-KDD. Workshop on Web mining and social network analysis. San Jose, California, pp 56–65
12. Milstein S, Chowdhury A, Hochmuth G, Lorica B, Magoulas R (2008) Twitter and the micro-blogging revolution. OReilly Radar report. http://weigend.com/files/teaching/haas/2009/readings/OReillyTwitterReport200811.pdf

13. Ozdikis O, Senkul P, Oguztuzun H (2012) Semantic expansion of Tweet contents for enhanced event detection in Twitter. In: International conference on advances in social networks analysis and mining (ASONAM), Istanbul, Turkey
14. Ozdikis O, Senkul P, Oguztuzun H (2012) Semantic expansion of hashtags for enhanced event detection in Twitter. Workshop on online social systems (WOSS). Istanbul, Turkey
15. Park H, Youn SB, Lee GY, Ko H (2011) Trendy episode detection at a very short time granularity for intelligent VOD service: a case study of live baseball game, In Proceedings of EuroITV
16. Pembe FC, Say ACC (2004) A Linguistically motivated information retrieval system for Turkish. In: Lecture notes in computer science, vol 3280, pp 741–750
17. Petrovic S, Osborne M, Lavrenko V (2010) Streaming first story detection with application to Twitter. In: Proceedings of the 11th conference of the North American chapter of the association for computational linguistics (NAACL HLT), Los Angeles, California
18. Plachouras V, Stavrakas Y (2012) Querying term associations and their temporal evolution in social data. Workshop on online social systems (WOSS). Istanbul, Turkey
19. Rapp R (2002) The computation of Word associations: comparing syntagmatic paradigmatic approaches. In: Proceedings of COLING, Taiwan
20. Rapp R (2004) A freely available automatically generated thesaurus of related words. In: Proceedings of 4th international conference on language resources and evaluation (LREC), Portugal
21. Sakaki T, Okazaki M, Matsuo Y (2010) Earthquake shakes Twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on World Wide Web, North Carolina, USA
22. Sankaranarayanan J, Samet H, Teitler BE, Lieberman MD, Sperling J (2009) TwitterStand: news in Tweets. In: Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems, Seattle, Washington, 04–06 Nov 2009
23. Sarma AD, Jain A, Yu C (2011) Dynamic relationship and event discovery. In: Proceedings of WSDM '11, pp 207–216
24. Sayyadi H, Hurst M, Maykov A (2009) Event detection and tracking in social streams. In: Proceedings of ICWSM 2009, San Jose CA, USA
25. Schutze H, Pedersen J (1993) A vector model for syntagmatic and paradigmatic relatedness. Making sense of words. In: Proceedings of the conference England, Oxford, pp 104–113
26. Song S, Li Q, Bao H (2012) Detecting dynamic association among Twitter topics. WWW 2012 poster presentation, Lyon, France, 16–20 Apr 2012
27. Song S, Li Q, Zheng N (2010) A spatio-temporal framework for related topic search in microblogging, In: Proceedings of the 6th international conference on active media technology, Toronto, Canada
28. Tan PN, Steinbach M, Kumar V (2006) Introduction to data mining. Addison Wesley, Reading, p 75
29. Wang M, Hsu P, Chuang YC (2011) Mining workflow outlier with a frequency-based algorithm. J Control Autom 4(2)

Springer

http://www.springer.com/978-3-319-05911-2

State of the Art Applications of Social Network Analysis

Can, F.; Özyer, T.; Polat, F. (Eds.)
2014, XII, 372 p. 137 illus., 94 illus. in color., Hardcover
ISBN: 978-3-319-05911-2