

Learning Similarity Metrics for Event Identification in Social Media

Hila Becker
Columbia University
hila@cs.columbia.edu

Mor Naaman
Rutgers University
mor@rutgers.edu

Luis Gravano
Columbia University
gravano@cs.columbia.edu

ABSTRACT

Social media sites (e.g., Flickr, YouTube, and Facebook) are a popular distribution outlet for users looking to share their experiences and interests on the Web. These sites host substantial amounts of user-contributed materials (e.g., photographs, videos, and textual content) for a wide variety of real-world events of different type and scale. By automatically identifying these events and their associated user-contributed social media documents, which is the focus of this paper, we can enable event browsing and search in state-of-the-art search engines. To address this problem, we exploit the rich “context” associated with social media content, including user-provided annotations (e.g., title, tags) and automatically generated information (e.g., content creation time). Using this rich context, which includes both textual and non-textual features, we can define appropriate document similarity metrics to enable online clustering of media to events. As a key contribution of this paper, we explore a variety of techniques for learning multi-feature similarity metrics for social media documents in a principled manner. We evaluate our techniques on large-scale, real-world datasets of event images from Flickr. Our evaluation results suggest that our approach identifies events, and their associated social media documents, more effectively than the state-of-the-art strategies on which we build.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms

Experimentation, Measurement

Keywords

Event Identification, Social Media, Similarity Metric Learning

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WSDM'10, February 4–6, 2010, New York City, New York, USA.
Copyright 2010 ACM 978-1-60558-889-6/10/02 ...\$10.00.

1. INTRODUCTION

The ease of publishing content on social media sites brings to the Web an ever increasing amount of content captured during—and associated with—real-world events. Sites like Flickr, YouTube, Facebook and others host user-contributed content for a wide variety of events. These range from widely known events, such as presidential inaugurations, to smaller, community-specific events, such as annual conventions and local gatherings. By automatically identifying these events and their associated user-contributed social media documents, which is the focus of this paper, we can enable powerful local event browsing and search, to complement and improve the local search tools that Web search engines provide. In this paper, we address the problem of how to identify events and their associated user-contributed documents over social media sites.

In one scenario, consider a person who is thinking of attending “All Points West,” an annual music festival that takes place in early August in Liberty State Park, New Jersey. Prior to purchasing a ticket, this person could search the Web for relevant information, to make an informed decision. Unfortunately, Web search results are far from revealing for this relatively minor event: the event’s website contains marketing materials, and traditional news coverage is low. Overall, these Web search results do not convey what this person should expect to experience at this event. In contrast, user-contributed content may provide a better representation of prior instances of the event from an attendee’s perspective. A user-centric perspective, as well as coverage of a wide span of events of varying type and scale, make social media sites a valuable source of event information.

Identifying events and their associated documents over social media sites is a challenging problem, as social media data is inherently noisy and heterogeneous. In our “All Points West” example, some photographs might contain the event’s name in the title, description, or tag fields, while many others might not be as clearly linked, with titles such as “Radiohead” or “Metric” and descriptions such as “my favorite band.” Photographs geo-tagged with the coordinates of Liberty State Park, and taken on August 8, 2008, are likely to be related to this event, regardless of their textual description, but not every photograph taken on August 8, 2008, or titled “Radiohead,” necessarily corresponds to this event. Overall, social media documents generally include information that is useful for identifying the associated events, if any, but this information is far from uniform in quality and might often be misleading or ambiguous.

Our problem is most similar to the event detection task [3, 26, 39], where the objective is to identify news events in a continuous stream of news documents (e.g., newswire, radio broadcast). However, our problem exhibits some fundamental differences from traditional event detection that originate from the focus on social media sources. Specifically, event detection traditionally aims to discover and cluster events found in textual news articles. These news articles adhere to certain grammatical, syntactical, and stylistic standards that are appropriate for their venue of publication. Therefore, most state-of-the-art event detection approaches leverage natural language processing tools such as named-entity extraction and part-of-speech tagging to enhance the document representation [19, 28, 40]. In contrast, social media documents contain little textual narrative, usually in the form of a short description, title, or keyword tags. Importantly, as discussed above, this text is often noisy, which renders traditional event detection techniques undesirable for social media documents, as we will see.

While social media documents present challenges for event detection, they also exhibit opportunities not found in traditional news articles. Specifically, social media documents have a wealth of associated “context,” including user-provided annotations (e.g., title, tags), and automatically generated information (e.g., upload or content creation time). Individual features might be noisy or unreliable, but collectively they provide revealing information about events, and this information is valuable to address our problem of focus.

In this paper, we exploit this rich family of features to identify events and their associated user-contributed social media documents. We explore distinctive representations of social media documents to analyze document similarity and identify which documents correspond to the same events. We define appropriate similarity metrics for each document representation, and explore a variety of techniques for combining them into a single measure of social media document similarity. We experiment with ensemble-based and classification-based similarity learning techniques, and use them in conjunction with a scalable, online clustering algorithm, to generate a clustering solution where each cluster corresponds to an event and includes the social media documents associated with the event.

The contributions of this paper are as follows:

- We pose the problem of identifying events and their user-contributed social media documents as a clustering task, where documents have multiple features, associated with domain-specific similarity metrics (Section 3).
- We propose a general online clustering framework, suitable for the social media domain (Section 4).
- We develop several techniques for learning a combination of the feature-specific similarity metrics, and use them to indicate social media document similarity in a general clustering framework¹ (Sections 5 and 6).
- We evaluate our proposed clustering framework and the similarity metric learning techniques on two real-world datasets of social media event content (Section 7).

We conclude with a discussion of the implications of our findings and directions for future work in Section 8.

¹One of these techniques was the focus of a preliminary, earlier workshop paper describing this work [6].

2. RELATED WORK

We describe relevant related work in four areas: large-scale data clustering, similarity metric learning, event detection and tracking in news streams, and social media analysis.

There are many approaches for clustering large-scale data [7], trading off runtime performance and clustering accuracy. One of the important issues to address when clustering large-scale data is how to compare the data elements against each other, which is hard to perform in a scalable manner as the size of the data grows.

Several solutions were proposed to alleviate this problem. One set of solutions [35, 41] uses statistical properties to represent subsets of the data, thus reducing the total number of comparisons to be made. In our work, we use this type of solution by representing clusters according to the average value of their elements. Other solutions propose “blocking” methods [9, 20, 30], which partition elements into several subsets based on a rough measure of similarity, and then use traditional clustering algorithms (e.g., K-means, EM [7]) on each subset, with exact similarities. We do not use blocking techniques in this paper due to the online setting of our problem, but plan to explore them in future work.

The choice of clustering similarity metric is critical for obtaining high-quality clustering solutions. In domains where more than one similarity metric is appropriate, several approaches have been proposed for combining multiple similarities using machine learning techniques [8, 10, 12, 13]. Other metric learning approaches use optimization techniques to learn a similarity metric from labeled examples directly [37, 14]. In our work, we define similarities tailored to the social media domain, and use classification-based and ensemble-based techniques to learn a combined similarity metric.

The topic detection and tracking (TDT) event detection task [2] was studied in a notable collective effort to discover and organize news events in a continuous stream (e.g., newswire, radio broadcast) [3, 26, 39]. With an abundance of well-formed text, many of the proposed approaches (e.g., [19, 40]) rely on natural language processing techniques to extract linguistically motivated features. Makkonen et al. [28] extracted meaningful semantic features such as names, time references, and locations, and learned a similarity function that combines these metrics into a single clustering solution. They concluded that augmenting documents with semantic terms did not improve performance, and reasoned that inadequate similarity functions were partially to blame. In our setting, clustering performance improves when we combine the variety of social media features judiciously.

Several efforts have focused on extracting high-quality information from social media [1, 4, 24, 27, 31]. Recent studies [21, 22] showed that social media document tags are accurate content descriptors, and could be used to train a social tagging prediction system. Tags have also been used in conjunction with other context [25] to retrieve Flickr images of geographical landmarks. Directly related to our problem, recent studies [11, 31] analyzed temporal and spatial tag distribution to identify tags corresponding to events. However, they did not attempt to aggregate related social media documents using the wealth of available context features.

3. PROBLEM DEFINITION

Given a set of social media documents associated with events, the problem that we address in this paper is how to

identify the events that are reflected in the documents (e.g., President Obama’s inauguration, or Madonna’s October 6, 2008 concert in Madison Square Garden), and to correctly assign the documents that correspond to each event. We cast our problem as a clustering problem over social media documents (e.g., photographs, videos, social network group pages), where each document includes a variety of “context features” with information about the document. Some of these features (e.g., title, description, tags) are manually provided by users, while other features (e.g., upload or content creation time) are automatically generated.

Problem Definition. Consider a set of social media documents where each document is associated with an (unknown) event. Our goal is to partition this set of documents into clusters such that each cluster corresponds to all documents that are associated with one event.

As the formal definition of “event,” we adopt the version used for the Topic Detection and Tracking (TDT) event detection task over broadcast news [38].

Definition. An event is something that occurs in a certain place at a certain time.

In our work, we make a couple of assumptions on the relationship between events and social media documents. First, we will consider documents that are significantly related to an event as being associated with the event, even if the documents were produced before or after the event. For instance, in our “All Points West” example, a photograph of a participant in front of the box office represents the author’s experience in the context of the event and will therefore be associated with the event for our purpose. Second, we assume that each social media document corresponds to exactly one event. However, our solution can easily be extended to handle cases where a single social media document contains information pertaining to several events.

As a distinctive characteristic, social media documents include a variety of *context features*, that are dependent on the type of document (e.g., a “duration” feature is meaningful for videos but not photographs). However, many social media sites share a core set of features. These features include: *author*, with an identifier of the user who created the document (e.g., “said&done” is the author of the photograph in Figure 1); *title*, with the “name” of the document (e.g., “DSC01325” in Figure 1); *description*, with a short paragraph summarizing the document contents (e.g., “radiohead performing” in Figure 1); *tags*, with a set of keywords describing the document contents (e.g., “apw, All, Points, West” in Figure 1); *time/date*, with the time and date when the document was published (e.g., August 9, 2008 in Figure 1);² *location*, with the location associated with the document (e.g., Jersey City, New Jersey in Figure 1). These context features, collectively, will prove helpful for capturing social media document similarity and, in turn, for identifying events and their associated documents, as we discuss next.

The context features of social media documents provide complementary cues for deciding when documents correspond to the same event. Individual features are often insufficient for this purpose, and all features collectively provide more reliable evidence. For example, the description of two

²Often documents include their capture or creation time (e.g., capture time/date, August 8, 2008 in Figure 1).

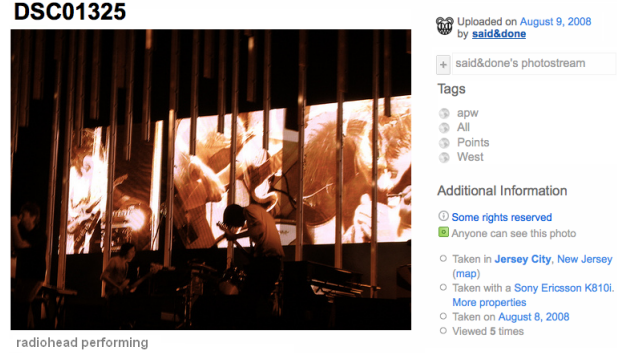


Figure 1: A Flickr photograph associated with the “All Points West” event.

images associated with the same event (e.g., the “All Points West” music festival) might be ambiguous or not very revealing (e.g., the description might read “my favorite band in concert” and “radiohead in concert”); but the images’ time/date and location (e.g., August 8, 2008, Liberty State Park, New Jersey) provide strong evidence that they are likely to be about the same event.

In this paper, we consider social media document representations using each individual feature, according to its type (e.g., textual or time data). In addition, we use one textual document representation that contains the textual representations of all the document features (title, description, tags, time/date and location). This representation, *all-text*, is commonly used in similar domains [28].

Next, we list the key *types* of features we extract from social media documents, and define individual similarity metrics for these feature types. It is possible, of course, to cluster the documents by using individual features according to an appropriate similarity metric. Such clustering approach is not ideal, since it does not exploit the wealth of context features collectively; instead, the rest of this paper describes strategies to consider the similarity metrics in concert.

Textual features: To exploit the various context features for our clustering task, we define a similarity metric for each feature, in a way that is appropriate for the feature’s domain. Specifically, we represent each textual feature (e.g., title, description, tags) as a *tf.idf* weight vector and use the cosine similarity metric, as defined in [26], as the feature similarity metric. We considered alternative *tf.idf* formulas such as Okapi [32]; however, they did not perform as well, so we do not discuss them further.

In addition, we considered traditional text processing steps such as stop-word elimination and stemming, and examined the effect of each of these with respect to the individual textual features. Instead of applying the same text processing treatment to all features, we conjectured that only some features would benefit from stemming or stop-word elimination. For instance, since tag keywords are meant to be a select set of descriptive keywords for the contents of the social media document, stop-word removal may not be appropriate (e.g., removing the tag “All” in our “All Points West” example). We empirically determined the appropriate stemming and stop-word settings for each textual feature (see Section 7.1).

Time/date: For time/date, an important feature in social media documents, we represent values as the number of min-

utes elapsed since the Unix epoch (i.e., since January 1st, 1970) and compute the similarity of two time/date values t_1 and t_2 as follows: if t_1 and t_2 are more than one year apart, we define their similarity as 0 (it is unlikely that the corresponding documents are associated with the same event in this case); otherwise, we define their similarity as $1 - \frac{|t_1 - t_2|}{y}$, where y is the number of minutes in a year.

Location: For location metadata associated with social media documents, we represent values as geographical coordinates (i.e., latitude-longitude pairs) and compute the similarity of two locations $\mathcal{L}_1 = (lat_1, long_1)$ and $\mathcal{L}_2 = (lat_2, long_2)$ as $1 - H(\mathcal{L}_1, \mathcal{L}_2)$, where $H(\cdot)$ is the Haversine distance [33], an accepted metric for geographical distance.

Having defined social media document representations and corresponding similarity metrics, we proceed to describe the general clustering framework in which they will be used.

4. CLUSTERING FRAMEWORK

We cast the problem of identifying events and their associated social media documents as a clustering problem. Ideally, each cluster should correspond to one event and consist of all of the social media documents associated with the event. In this section, we discuss the choice of general clustering algorithm for our scenario. Later, in Sections 5 and 6, we describe the key technical challenge of choosing a similarity metric for the clustering algorithm.

4.1 Scalable Clustering Approach

For our social media document scenario, the clustering algorithm of choice should be scalable, to handle the large volume of data in social media sites, and not require *a priori* knowledge of the number of clusters, since social media sites are constantly evolving and growing in size. Therefore, traditional clustering approaches that require knowledge of the number of clusters, such as K-means and EM [7], are not suitable for this problem. Other alternatives such as scalable graph partitioning algorithms [23] do not capture the highly skewed event distribution of social media event data due to their bias towards balanced partitioning (we experimented with graph partitioning algorithms, but do not discuss their results here because of their poor performance for our task).

Threshold-based techniques are preferable for our clustering task since they can be tuned using a training set and subsequently generalized to unseen data points. Hierarchical clustering algorithms [7], while relying on threshold tuning, are also not appropriate since they require processing a fully specified similarity matrix, which does not scale to the large size of our data. Furthermore, online or incremental clustering algorithms, which are able to handle a constant stream of new documents, are also desirable in our setting, where new documents are continuously being produced.

Based on these observations, we propose using a single-pass incremental clustering algorithm with a threshold parameter that can be tuned in a principled manner during a training phase. Single-pass incremental clustering has been shown to be an effective technique for event detection in textual news documents (e.g., [3, 39]). Such a clustering algorithm considers each element in turn, and determines the suitable cluster assignment based on the element’s similarity to any existing clusters. Specifically, given a threshold μ , a similarity function σ , and documents to cluster d_1, \dots, d_n , the algorithm considers each document d_i in order, and computes its similarity $\sigma(d_i, c_j)$ against each existing cluster c_j ,

for $j = 1, \dots, k$. (Initially, $k = 0$.) Different versions of the algorithms differ on how this similarity σ is computed, as we report in the next section. If there is no cluster whose similarity to d_i is greater than μ , we increment k by one and create a new cluster c_k for d_i . Otherwise, d_i is assigned to a cluster c_j with maximum $\sigma(d_i, c_j)$.

Conceptually, the similarity $\sigma(d, c)$ between a document d and a cluster c can be computed by comparing the features of d to those of the cluster c ; or by directly comparing d to the documents in cluster c . We propose methods that use both approaches. In Section 5.2, we describe a simple similarity approach, comparing d to every document in the cluster c , and define $\sigma(d, c)$ as the average similarity score, for a suitable document similarity metric. In other words, we can define $\sigma(d, c) = \sum_{d' \in c} \frac{\sigma(d, d')}{|c|}$. This approach is not efficient because it requires comparing document d against every document in cluster c .

A more efficient approach is to represent each cluster using the centroid of its documents. The centroid for a cluster of documents c is defined as $\frac{1}{|c|} \sum_{d \in c} d$. Depending on the document representation we use, our centroids are either the average *tf.idf* score per term (for textual features such as title, description, tags), the average time in minutes (for time/date), or the geographic mid-point (for location) of all documents in c . We use the centroid similarity approach in the majority of our techniques, described in detail in Sections 5.3 and 6.

4.2 Quality Metrics and Thresholding

Regardless of the definition of $\sigma(d, c)$, the clustering algorithm on which we focus requires that we specify a clustering threshold μ . To tune the clustering threshold for a specific dataset, we run the clustering algorithm on a subset of labeled training data. We evaluate the algorithm’s performance on the training data using a range of thresholds, and identify the threshold setting that yields the highest-quality solution according to a given clustering quality metric. Although several clustering quality metrics exist (see [5]), in this paper we focus on Normalized Mutual Information (NMI) [29, 34] and B-Cubed [5]. Both NMI and B-Cubed balance our desired clustering properties: maximizing the homogeneity of events within each cluster, and minimizing the number of clusters that documents for each event are spread across.

NMI is an information-theoretic metric that was originally proposed as the objective function for cluster ensembles [34]. NMI measures how much information is shared between actual “ground truth” events, each with an associated document set, and the clustering assignment. Specifically, for a set of clusters $C = \{c_1, \dots, c_J\}$ and events $E = \{e_1, \dots, e_K\}$, where each c_j and e_k is a set of documents, and n is the total number of documents, $NMI(C, E) = \frac{I(C, E)}{(H(C) + H(E))/2}$, where $I(C, E) = \sum_k \sum_j \frac{|e_k \cap c_j|}{n} \log \frac{n \cdot |e_k \cap c_j|}{|e_k| \cdot |c_j|}$, $H(C) = - \sum_j \frac{|c_j|}{n} \log \frac{|c_j|}{n}$, and $H(E) = - \sum_k \frac{|e_k|}{n} \log \frac{|e_k|}{n}$.

B-Cubed estimates the precision and recall associated with each document in the dataset individually, and then uses the average precision P_b and average recall R_b values for the dataset to compute $B-Cubed = \frac{2 \cdot P_b \cdot R_b}{P_b + R_b}$. For each document, precision is defined as the proportion of items in the document’s cluster that correspond to the same event, and recall is defined as the proportion of documents that correspond to the same event, which are also in the document’s cluster.

As we mentioned, the choice of clustering quality metric serves an important role in our clustering approach since it is used to tune the threshold parameter μ . Although NMI and B-Cubed capture the clustering properties that we are interested in, it is not always the case that the best threshold setting according to NMI is also the best setting according to B-Cubed. In order to select the threshold setting that optimizes both metrics, we use a single aggregate objective function, equally weighing NMI and B-Cubed. The threshold setting that yields the highest combined NMI and B-Cubed value is considered Pareto optimal [16], meaning that we cannot find a threshold with higher NMI value that does not have a lower B-Cubed value and vice versa.

The general clustering algorithm that we described relies heavily on a similarity metric σ for two documents, or for a document and a cluster centroid. In the next section, we turn to the crucial issue of learning such a similarity metric.

5. ENSEMBLE-BASED SIMILARITY

Our first attempt at learning a similarity metric using the wealth of context features present in social media documents involves an ensemble algorithm, which considers each feature as a weak indication of social media document similarity, and combines all features using a weighted similarity consensus function. Ensemble clustering is an approach that combines multiple clustering solutions for a document set [17, 18, 34]. The advantage of using an ensemble approach is its ability to account for different similarity metrics during the clustering process, by learning their optimal weighted contribution to the final clustering decision. In this section, we discuss ensemble clustering and show how we use it in conjunction with our clustering framework from Section 4 to learn a similarity metric for social media documents.

5.1 Training a Cluster Ensemble

The first step in any ensemble clustering approach is to select techniques for partitioning the data. These techniques, also referred to as *clusterers* (C_1, \dots, C_m in Figure 2(b)), produce mappings from documents to clusters. Each of these techniques should have a unique view of the data (R_1, \dots, R_m in Figure 2(a)), or use a different underlying model to generate the data partitions. For our ensemble, we select clusterers that partition the data using the different social media features and appropriate similarity metrics discussed in Section 3. In particular, we have separate clusterers for features such as title, description, tags, location, and time. Following the logic of Section 4, we use the single-pass incremental clustering algorithm for each feature individually, with its respective similarity metric from Section 3, as the clustering similarity function σ . We tune the threshold μ for each clusterer on a set of training data, and select the best threshold based on each clusterer’s performance according to NMI and B-Cubed (see Section 4). This results in clusterers C_1, \dots, C_m (Figure 2(b)).

The clustering quality metrics described in Section 4 serve two important purposes in our ensemble approach. The first, as previously mentioned, is to select the most suitable threshold setting for each clusterer. The second is to assign a weight to each clusterer, indicating our confidence in its predictions. The weights are assigned during a supervised training phase, and used to determine each clusterer’s influence on the overall ensemble similarity assignment. By assigning a weight to a clusterer, we indicate how success-

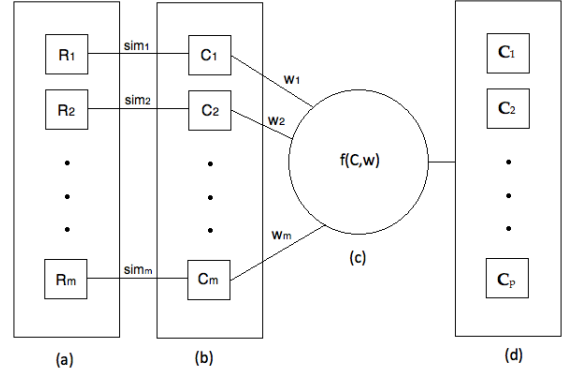


Figure 2: A conceptual diagram of an ensemble clustering process.

ful the clusterer was in capturing document similarity on a training set, and therefore how likely it is to correctly indicate the similarity of unseen document pairs.

Once we select the best performing thresholds for all clusterers C_1, \dots, C_m , we set their weights w_1, \dots, w_m to equal their respective combined NMI and B-Cubed scores (see Section 4), and then normalize the ensemble weights such that $\sum_{i=1}^m w_i = 1$. In the conclusion of the ensemble training phase, we have learned an optimal threshold for each clusterer, as well as a quality measure that will be used to weigh its decisions. With this information, we can proceed in two distinct ways: the first is to combine individual clusterer partitions as in the traditional ensemble clustering setting (Section 5.2), and the second is to use the learned weights and thresholds as a model for the similarity metric, without further influence from the individual clusterers (Section 5.3). We elaborate on these approaches next.

5.2 Combining Individual Partitions

The first ensemble-based approach for learning a similarity metric follows the traditional cluster ensemble framework [34] that utilizes individual clusterers’ similarity judgements on document pairs. Given a set of documents, we use each clusterer with its learned threshold to generate a clustering partition. Our challenge is to develop a consensus mechanism for combining these individual partitions into one clustering solution (C_1, \dots, C_p in Figure 2(d)). The consensus that our algorithm reaches using the clusterers’ similarity judgements is translated into a similarity metric σ that can be used in our general clustering framework (Section 4).

Intuitively, each clusterer can be regarded as providing an expert vote on whether two documents belong in the same cluster. The consensus function we use is a weighted binary vote: for a pair of documents (d_i, d_j) and clusterer C , we define a prediction function $P_C(d_i, d_j)$ as equal to 1, if d_i and d_j are in the same cluster, or 0 otherwise³. Then, we compute the consensus score for d_i and d_j as $\sum_C P_C(d_i, d_j) \cdot w_C$, where w_C is the weight of clusterer C . For example, consider a simple ensemble with three clusterers $C_{\text{time}}, C_{\text{location}}$, and C_{tags} , whose weights are 0.25, 0.35, and 0.4, respectively. To determine whether two documents d_i and d_j belong in the same cluster, we compute their prediction $P_{C_i}(d_i, d_j)$,

³Similarly, we can use the raw similarity score.

for $i = \text{time}, \text{location}, \text{and tags}$. Suppose that C_{time} and C_{location} cluster d_i and d_j together, but C_{tags} does not. The consensus score for d_i and d_j is then $0.25+0.35=0.6$.

Note that our general single-pass incremental clustering algorithm has to compare each document to existing *clusters* at every step. However, in the cluster ensemble formulation we can only obtain the clusterers’ similarity judgements for *document* pairs. Therefore, in order to measure the similarity of a document to a cluster, we compare the document against all documents in the cluster using the ensemble consensus function, and use the average consensus score as our similarity metric σ for this document-cluster pair.

Learning a similarity metric using this ensemble approach yields a simple model, which uses a weighted combination of the data partitions obtained by clustering according to each feature and corresponding similarity metric from Section 3. While this approach provides an intuitive solution that models the contribution of each feature-specific similarity in a clustering context, one of its main drawbacks is its best-case quadratic running time in the size of the dataset. In the next section we therefore consider a modified approach that still uses the knowledge from the ensemble training phase to combine the similarity metrics, while at the same time improves efficiency with a centroid-based similarity technique.

5.3 Combining Individual Similarities

The second ensemble-based technique for learning a similarity metric uses the threshold and weight assignment learned in the ensemble training phase (Section 5.1) as the only input from the clusterers. Instead of computing the consensus score using the clusterers’ predictions, we now compute the documents’ feature-specific similarity metrics directly for documents and cluster centroids. The advantages of this modification to the ensemble similarity learning technique include improved efficiency via the use of centroids, providing for a more direct similarity metric computation.

To compute a similarity between a document d_i and a cluster centroid c_j , we repeat the same decision procedure for the similarity of document pairs, described above, using the weight and threshold that we learned for each individual feature. For similarity metric σ_C , threshold μ_C , and weight w_C associated with a clusterer C , we define $P_C(d_i, c_j) = 1$ if $\sigma_C(d_i, c_j) > \mu_C$, and 0 otherwise, and compute the combined similarity metric $\sum_C P_C(d_i, c_j) \cdot w_C$. Note that while this formulation of the similarity function uses a weighted binary vote for each feature, we could alternatively use the raw similarity score, as we suggest in the next section.

Note that we can now use the one-pass incremental clustering algorithm with centroid similarity. Depending on the document representation, the centroid is either the average *tf.idf* score per term (for textual features such as title, description, tags), the average time in minutes (for time/date), or the geographic mid-point (for location). Centroids can be updated and maintained with little cost using the general framework described in Section 4.

6. CLASSIFICATION-BASED SIMILARITY

In this section, we use classification models to learn document similarity functions for social media, as an alternative to the ensemble-based approach. In other words, we use a classifier with similarity scores as features to predict whether a pair of documents belongs to the same event. Formally, given a pair of social media documents d_i and d_j , we

compute the raw similarity scores $\sigma_1(d_i, d_j), \dots, \sigma_m(d_i, d_j)$, corresponding to the document features and individual similarity metrics defined in Section 3. Using this formulation of the problem, we are able to utilize a variety of state-of-the-art classification algorithms for learning the combined similarity metric σ for our general clustering framework.

Before we can train a similarity metric classifier, we must decide whether to model similarity between document pairs, or documents-centroid pairs. Although we are interested in learning a similarity metric that would indicate when social media documents correspond to the same event, in our clustering framework we compare documents to cluster centroids. Therefore, we consider the alternative of training the classifiers on document-centroid pairs, which more closely resembles the data that the classifier will be predicting on.

Intuitively, modeling the similarity between documents and centroids would be more robust than modeling similarities between document pairs. For example, consider a pair of documents that does not share any tag keywords, yet relates to the same event. Having this pair as a positive example (i.e., the documents are about the same event) provides a false indication that tag keywords do not contribute towards a positive prediction. For centroids, since we aggregate and average the *tf.idf* values of multiple documents, there exists a better chance to capture some overlapping tag vocabulary and therefore to more accurately gauge the contribution of tag keywords to the overall similarity metric.

One key challenge for the classification-based approach involves the selection of training examples from which to learn the similarity classifiers. Ideally, we want our model to correctly predict the similarity of every document to every other document (or every centroid, based on the modeling choice described above) in the dataset. However, creating a training example for each document (or document-centroid) pair results in a skewed label distribution, since a large majority of pairs in the training dataset do not belong to the same event. Using a classifier trained with a skewed label distribution as a similarity metric for clustering yields poor clustering solutions since this classifier is much more likely to predict that two items do not belong in the same cluster, thus splitting single events across many clusters.

With this in mind, we can outline two sampling strategies to balance the label distribution. The first strategy is to take the first n documents in the training set according to their upload time, and compare them to every other document in that set. In the case of document-centroid similarities, we compare each document against all centroids, which are computed in advance for each event. To handle the skewed label distribution, we produce a random subsample of this data such that the number of positive and negative examples is balanced. We empirically found that generating a subsample that is 10% of the original sample size, with a balanced label distribution, yields a more accurate similarity metric classifier than other sampling techniques that we experimented with.

The second strategy is to select documents at random, pairing each document with one positive example, randomly selected from the set of documents that share the same event, and one negative example, randomly selected from the set of documents related to different events. For document-centroid pairs, we only have one choice for the positive example per document, but we randomly select among different event centroids for the negative document-centroid pair.

For this family of similarity metric learning techniques, we consider a variety of state-of-the-art classification algorithms, and train them using the datasets discussed in this section. We elaborate on our choice of classifiers and the training process in the next section.

7. EXPERIMENTS

We evaluated our work on a large dataset of real world data from popular social media sites, with these goals:

- Examine which sampling and modeling methods, and what classification algorithms perform well for the classification-based approach.
- Determine which similarity metrics and techniques perform best for the event identification task.
- Gain insight about these approaches by analyzing the weights that the similarity metric learning approaches assign to each feature-specific similarity.

We report on the dataset and experimental settings, then turn to the results of our experiments.

7.1 Experimental Settings

Data: For our experiments, we collected two datasets of labeled event photographs from Flickr, a popular photo-sharing service, using the site’s API⁴. The *Upcoming* dataset consists of all photographs that were manually tagged by users with an event id corresponding to an event from the Upcoming event database⁵. These Upcoming tags provide the “ground truth” for our clustering experiments (see Section 4). Each photograph corresponds to a single event, and each event is self-contained and independent of other events in the dataset. The *Upcoming* dataset contains 9,515 unique events, with an average of 28.42 photographs per event, for a total of 270,425 photographs, taken between January 1, 2006, and December 31, 2008.

Our second dataset is the *Last.fm* dataset, which consists of all Flickr photographs that were manually tagged by users with an id corresponding to an event from the Last.fm music event catalog⁶. The *Last.fm* dataset contains 24,958 unique events, with an average of 23.84 photographs per event, for a total of 594,946 photographs, taken between January 1, 2006, and December 31, 2008.

The context features associated with each photograph include the title, description, tags, time/date of capture, and location. On average, 32.2% of the photos include location information in the form of geo-coordinates. On this subset of the data, we perform reverse geo-coding using the Flickr API, to obtain a textual representation of the location of each photo, which we use for the *all-text* feature.

Training Methodology: We train our clustering algorithms on data from the *Upcoming* dataset, and test them on unseen *Upcoming* data, as well as *Last.fm* data. We order the photographs in the *Upcoming* dataset according to their upload time, and then divide them into three equal parts. We use the earliest two thirds of the data as training and validation sets. We use the *training set* to tune the clusterer thresholds for the ensemble-based techniques and train classifiers for the classification-based techniques. We use the *validation set* to learn the weights for the ensemble and tune

the threshold for the general single-pass incremental clustering algorithm. The last third of the *Upcoming* data and all of the *Last.fm* data are used as *test sets*, on which we report our results. We chose a time-based split since it best emulates real-world scenarios, where we only have access to past data with which we can train models to cluster future data. We train our similarity metrics once and for all, without adapting them as we observe more data. Dynamically modifying the similarity metrics as new documents arrive is reserved for future work.

Document Representations: The Lemur Toolkit⁷ is used to index our documents according to each textual representation discussed in Section 3. These representations include *Title*, *Tags*, *Description*, and *All-Text*. We use all possible settings of stemming and stop-word elimination for each document representation, and create a separate index for every possible combination. We use the index to compute *tf.idf* vectors for each textual document representation. Finally, we create additional document representations using numeric time/date (*Time/Date-Proximity*) and location coordinates (*Location-Proximity*) as described in Section 3. If a document representation cannot be created due to missing data (e.g., an unspecified location), we assign it a similarity value of 0 to any other document for this representation.

Weighting Clusterers: For the ensemble-based approaches, we use Lemur’s single-pass incremental clustering implementation to cluster the training data according to each document representation and corresponding similarity metric from Section 3. We tune the clustering threshold for each clusterer using the training set, considering thresholds in the range [0, 1], with 0.05 increments. For time and location features, we apply log scaling to the similarity metric in order to perform the selection of thresholds with a finer granularity, as appropriate to those metrics. For each document representation, we select the threshold that yields the highest combined NMI and B-Cubed score (Section 4). For textual document representations, we select one threshold setting per feature and associated parameter settings (stemming and stop-word elimination). We use the best-performing setting for each textual representation when creating future document representations for that feature. The best settings for *Title* and *Description* were no stemming or stop-word elimination, while *Tags* benefited from stemming and *All-Text* from stop-word elimination.

We proceed to cluster the validation set according to each document representation and corresponding similarity metric, using the selected threshold setting for each clusterer. To determine the weight of each clusterer, we compute its combined NMI and B-Cubed scores on the validation set. Finally, we run the ensemble algorithm on the validation set using the selected clusterers, and tune the clustering threshold for the ensemble approach using NMI and B-Cubed.

Training Classifiers: To train similarity classification models (Section 6), we used the training set to construct four training samples according to the modeling and sampling strategies that we discussed in Section 6:

- TIME-DD: all possible document-document pairs from the first 500 documents ordered according to their time of creation.
- RANDOM-DD: 10,000 document-document pairs chosen randomly from all possible pairings between documents.

⁴<http://www.flickr.com/services/api>

⁵<http://www.upcoming.org>

⁶<http://www.last.fm/events>

⁷<http://www.lemurproject.org>

- TIME-DC: all possible document-centroid pairs from the first 500 documents, ordered according to their time of creation, and their corresponding centroids.
- RANDOM-DC: 10,000 document-centroid pairs chosen randomly from all possible pairings between documents and centroids.

For the document-centroid modeling approach, we computed all event centroids based on the ground truth labels.

We used the Weka toolkit [36] to build classifiers for all of the above training sets. We explored a variety of classifier types and selected two techniques that yielded the best overall performance in preliminary tests using the training set, although differences were not substantial. We selected support vector machines (Weka’s sequential minimal optimization implementation), and logistic regression.

Comparing Techniques: We consider all individual clusterers as baseline approaches, namely, *All-Text*, *Title*, *Description*, *Tags*, *Time/Date-Proximity*, and *Location-Proximity*. We compared them against our clustering approaches using four different similarity metric learning techniques:

- ENS-PART: Ensemble-based approach, combining partitions (Section 5.2).
- ENS-SIM: Ensemble-based approach, combining similarity scores (Section 5.3).
- CLASS-SVM: Similarity classifier, using Support Vector Machines (Section 6).
- CLASS-LR: Similarity classifier, using Logistic Regression (Section 6).

To evaluate the clustering solutions of these different techniques, we use the clustering quality metrics of Section 4, namely, NMI and B-Cubed.

7.2 Experimental Results

We begin with the task of finding the best modeling and sampling strategies for the classification-based techniques, which is of course critical for the performance of these approaches. We trained a classifier using support vector machines and logistic regression for the different sampling and modeling strategies, and tested the quality of clustering results for each classifier and sampling method. The results are shown in Table 1, indicating that time-based sampling is consistently superior to random sampling according to both NMI and B-Cubed. Similarly, the document-centroid modeling techniques yield higher-quality clustering solutions than techniques that model similarity between document pairs. We therefore proceed to test our classification-based techniques using classifiers trained on the time-based document-centroid training sample (TIME-DC).

Next, we compared our similarity metric learning techniques against each other, as well as against the top performing individual clusterers, on the *Upcoming* test set. Table 2 presents the clustering performance of all similarity metric learning techniques, as well as the *All-Text* and *Tags* clusterers, in terms of NMI and B-Cubed. Not surprisingly, the top performing *individual* clusterer is *All-Text*.

More importantly, the similarity metric combination approaches that we consider in this work outperform all individual clusterers, including *All-Text* (which also considers all document features, but with a single text-based similarity metric). Among the similarity metric learning tech-

Algorithm	Sample	NMI	B-Cubed
CLASS-SVM	TIME-DC	0.9492	0.8226
CLASS-SVM	TIME-DD	0.9396	0.7868
CLASS-SVM	RANDOM-DC	0.9082	0.6954
CLASS-SVM	RANDOM-DD	0.8227	0.4180
CLASS-LR	TIME-DC	0.9508	0.8258
CLASS-LR	TIME-DD	0.9360	0.7743
CLASS-LR	RANDOM-DC	0.8991	0.6483
CLASS-LR	RANDOM-DD	0.8257	0.4360

Table 1: Performance of classification-based techniques using different sampling strategies over the validation set.

Algorithm	NMI	B-Cubed
All-Text	0.9240	0.7697
Tags	0.9229	0.7676
ENS-PART	0.9296	0.7819
ENS-SIM	0.9322	0.7861
CLASS-SVM	0.9425	0.8095
CLASS-LR	0.9444	0.8155

Table 2: Performance of all similarity metric learning techniques and the best individual clustering techniques over the *Upcoming* test set.

niques, the classification-based techniques CLASS-SVM and CLASS-LR outperform the ensemble-based techniques ENS-PART and ENS-SIM. CLASS-LR is the best overall technique in terms of both NMI and B-Cubed. The least successful of our techniques is ENS-PART, implying that learning the similarity metric directly is more effective than combining individual feature-based clustering partitions. Some events identified by CLASS-LR are shown in Table 3.

We also compared our techniques using the *Last.fm* dataset as an independent test set (with the training and validation set from the *Upcoming* dataset). As Figure 3 shows, the test on the *Last.fm* dataset resulted in similar, albeit not identical, outcomes. In that test, all similarity metric learning techniques still outperform the baselines, but the top-performing technique is now ENS-SIM. Recall that the analysis of our techniques is performed over data from Flickr, with one dataset containing content annotated with events from Upcoming, and the other from Last.fm. Different properties of Last.fm events compared to Upcoming events could be the source of these relative performance differences (e.g., *Tags* similarity is better than *All-Text* for the *Last.fm* dataset), in which case ENS-SIM may be most robust in the face of these differences. Interestingly, the strong results for all methods over *Last.fm* are encouraging, as some real-world scenarios will require training on datasets different than the eventual data to be analyzed.

To determine if our results are statistically significant, we executed a set of tests by partitioning the *Upcoming* test dataset into 10 equal subsets according to document upload time, and ran each clustering technique on every subset. We discuss detailed results only for the NMI metric (while

Title	Date	Location	#Docs
Street Art Photowalk	7/14/08	London	411
Cherry Blossom Festival	4/12/08	San Francisco	269
American Music Union	8/8/08	Pittsburgh	209
How Weird Street Fair	5/4/08	San Francisco	52

Table 3: Some events identified by CLASS-LR.



Figure 3: NMI and B-Cubed scores on the *Upcoming* and *Last.fm* test datasets for All-Text (A), Tags (B), ENS-PART (C), ENS-SIM (D), CLASS-SVM (E), and CLASS-LR (F).

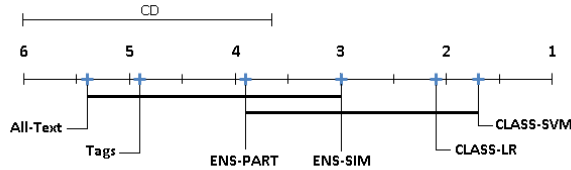


Figure 4: Comparison of all techniques using the Nemenyi test. Groups of techniques connected by a line are not significantly different at $p < 0.05$.

trends for B-Cubed were equivalent to trends observed for NMI, the differences between approaches as measured by B-Cubed were not as significant). We used the Friedman test [15], a non-parametric statistical test for comparing a set of alternative models. The Friedman test’s null hypothesis states that all the approaches have similar performance. The results of the test comparing the 10 runs show that we can reject this null hypothesis with $p < 0.05$, meaning that the performance of some approaches is significantly different.

A post-hoc statistical test is required to expose the relationship between the individual techniques. Figure 4 shows the results of the post-hoc analysis of our data using the Nemenyi test and the graphical representation as proposed by Demšar to visualize the relationships between the techniques [15]. Techniques are plotted according to their average rank for the test datasets, and a line spans each group of techniques that is not different in a statistically significant manner. The figure demonstrates that, for the 10 tests, while CLASS-SVM and CLASS-LR are significantly better than both baseline approaches, they are not significantly different from each other, or the other similarity metric learning techniques, at the $p < 0.05$ level. For $p < 0.1$, we can claim that CLASS-SVM is also significantly better than ENS-PART.

To gain more insight into the results of the various techniques, we analyzed the similarity metric models. Since the techniques use different modeling assumptions, we examined their differences in terms of the weight coefficients that they assign to each similarity feature. These coefficients, while not comparable in absolute terms, hint at the relative contribution of each similarity feature towards the model’s final similarity prediction. CLASS-LR considers *All-Text* as the

most important feature, followed by *Time/Date-Proximity*. CLASS-SVM, on the other hand, considers *Title*, followed by *All-Text* as the top two features. A surprising result is that both classifiers agree that, in the presence of all other features, *Location-Proximity* is an indication of document dissimilarity. In contrast, our ensemble model gives the lowest weights to *Title* and *Time/Date-Proximity*, and *Location-Proximity* has the third highest weight (after *Tags* and *All-Text*). These observations can form the basis of a more detailed analysis in the future.

8. CONCLUSIONS

In this paper, we presented several novel techniques for identifying events and their associated social media documents, by combining multiple context features of the document in a variety of disciplined ways. We proposed a general framework for identifying events in social media documents via clustering, and used similarity metric learning approaches in this framework, to produce high quality clustering results. We discussed and experimented with ensemble-based and classification-based techniques, tailored to the social media domain, for combining a set of similarity metrics to predict when social media documents correspond to the same event. Our experiments suggest that our similarity metric learning techniques yield better performance than the baselines on which we build. In particular, our classification-based techniques show significant improvement over traditional approaches that use text-based similarity.

As the amount of social media content grows, research will have to identify robust ways to organize and filter that content. We provided a first step toward organizing media from real-life events. In future work, we will learn to distinguish between event and non-event documents (our current work focuses on event documents only). Other future directions include learning to rank events (e.g., to decide which events to feature in a browsing application), and presentation and summarization of event content [24].

9. ACKNOWLEDGMENTS

This material is based upon work supported by a generous research award from Google and by the National Science Foundation under Grants CNS-0717544 and IIS-0811038. We also thank Luis Alonso, Krzysztof Czuba, and Julia Stoyanovich for their feedback on our work.

10. REFERENCES

- [1] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. Finding high-quality content in social media. In *Proceedings of the First ACM International Conference on Web Search and Data Mining (WSDM’08)*, 2008.
- [2] J. Allan. Introduction to topic detection and tracking. In J. Allan, editor, *Topic Detection and Tracking – Event-based Information Organization*, pages 1–16. Kluwer Academic Publisher, 2002.
- [3] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *Proceedings of the 21st ACM International Conference on Research and Development in Information Retrieval (SIGIR’98)*, 1998.
- [4] S. Amer-Yahia, M. Benedikt, L. V. S. Lakshmanan, and J. Stoyanovich. Efficient network aware search in collaborative tagging sites. *PVLDB*, 1(1):710–721, 2008.
- [5] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 2008.

- [6] H. Becker, M. Naaman, and L. Gravano. Event identification in social media. In *Proceedings of the ACM SIGMOD Workshop on the Web and Databases (WebDB '09)*, June 2009.
- [7] P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.
- [8] M. Bilenko, S. Basu, and M. Sahami. Adaptive product normalization: Using online learning for record linkage in comparison shopping. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM'05)*, 2005.
- [9] M. Bilenko, B. Kamath, and R. J. Mooney. Adaptive blocking: Learning to scale up record linkage. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM'06)*, 2006.
- [10] M. Bilenko and R. J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*, 2003.
- [11] L. Chen and A. Roy. Event detection from Flickr data through wavelet-based spatial analysis. In *Proceedings of the 2009 ACM CIKM International Conference on Information and Knowledge Management (CIKM '09)*, 2009.
- [12] Z. S. Chen, D. V. Kalashnikov, and S. Mehrotra. Exploiting context analysis for combining multiple entity resolution systems. In *Proceedings of the 2009 ACM International Conference on Management of Data (SIGMOD'09)*, 2009.
- [13] W. W. Cohen and J. Richman. Learning to match and cluster large high-dimensional data sets for data integration. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*, 2002.
- [14] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th International Conference on Machine Learning (ICML'07)*, 2007.
- [15] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.
- [16] U. M. Diwekar. *Introduction to applied optimization*. Springer, 2003.
- [17] C. Domeniconi and M. Al-Razgan. Weighted cluster ensembles: Methods and analysis. *ACM Transactions on Knowledge Discovery from Data*, 2(4):1–40, 2009.
- [18] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. In *Proceedings of the 21st International Conference on Data Engineering (ICDE'05)*, April 2005.
- [19] V. Hatzivassiloglou, L. Gravano, and A. Maganti. An investigation of linguistic features and clustering algorithms for topical document clustering. In *Proceedings of the 23rd ACM International Conference on Research and Development in Information Retrieval (SIGIR'00)*, 2000.
- [20] M. A. Hernández and S. J. Stolfo. The merge/purge problem for large databases. In *Proceedings of the 1996 ACM International Conference on Management of Data (SIGMOD'96)*, 1995.
- [21] P. Heymann, G. Koutrika, and H. Garcia-Molina. Can social bookmarking improve web search? In *Proceedings of the First ACM International Conference on Web Search and Data Mining (WSDM'08)*, Feb. 2008.
- [22] P. Heymann, D. Ramage, and H. Garcia-Molina. Social tag prediction. In *Proceedings of the 31st ACM International Conference on Research and Development in Information Retrieval (SIGIR'08)*, July 2008.
- [23] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning: Application in VLSI domain. In *Proceedings of the 34th ACM Conference on Design Automation (DAC'97)*, 1997.
- [24] L. Kennedy and M. Naaman. Less talk, more rock: Automated organization of community-contributed collections of concert videos. In *Proceedings of the 18th International World Wide Web Conference (WWW'09)*, 2009.
- [25] L. Kennedy, M. Naaman, S. Ahern, R. Nair, and T. Rattenbury. How Flickr helps us make sense of the world: context and content in community-contributed media collections. In *Proceedings of the 15th International Conference on Multimedia (MULTIMEDIA'07)*, 2007.
- [26] G. Kumaran and J. Allan. Text classification and named entities for new event detection. In *Proceedings of the 27th ACM International Conference on Research and Development in Information Retrieval (SIGIR'04)*, 2004.
- [27] L. Liu, L. Sun, Y. Rui, Y. Shi, and S. Yang. Web video topic discovery and tracking via bipartite graph reinforcement model. In *Proceedings of the 17th International World Wide Web Conference (WWW'08)*, 2008.
- [28] J. Makkonen, H. Ahonen-Myka, and M. Salmenkivi. Simple semantics in topic detection and tracking. *Information Retrieval*, 7(3–4):347–368, 2004.
- [29] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge Univ. Press, 2008.
- [30] A. McCallum, K. Nigam, and L. H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'00)*, 2000.
- [31] T. Rattenbury, N. Good, and M. Naaman. Towards automatic extraction of event and place semantics from Flickr tags. In *Proceedings of the 30th ACM International Conference on Research and Development in Information Retrieval (SIGIR'07)*, pages 103–110, 2007.
- [32] S. E. Robertson and S. Walker. Okapi/Keenbow at TREC-8. In *Proceedings of the Fourteenth Text REtrieval Conference (TREC-8)*, 1999.
- [33] R. W. Sinnott. Virtues of the Haversine. *Sky and Telescope*, 68:159, 1984.
- [34] A. Strehl, J. Ghosh, and C. Cardie. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.
- [35] S. C. A. Thomopoulos, D. K. Bougoulas, and C.-D. Wann. Dignet: an unsupervised-learning clustering algorithm for clustering and data fusion. *IEEE Transactions on Aerospace Electronic Systems*, 31:21–38, Jan. 1995.
- [36] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2 edition, 2005.
- [37] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, 2002.
- [38] Y. Yang, J. Carbonell, R. Brown, T. Pierce, B. T. Archibald, and X. Liu. Learning approaches for detecting and tracking news events. *IEEE Intelligent Systems Special Issue on Applications of Intelligent Information Retrieval*, 14(4):32 – 43, 1999.
- [39] Y. Yang, T. Pierce, and J. Carbonell. A study on retrospective and on-line event detection. In *Proceedings of the 21st ACM International Conference on Research and Development in Information Retrieval (SIGIR'98)*, 1998.
- [40] K. Zhang, J. Zi, and L. G. Wu. New event detection based on indexing-tree and named entity. In *Proceedings of the 30th ACM International Conference on Research and Development in Information Retrieval (SIGIR'07)*, 2007.
- [41] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM International Conference on Management of Data (SIGMOD'96)*, 1996.