

Flexible Services

Social Video Event Discovery by Clustering

Document Name:	Clustering Test Report
Project Title:	Social Video
Document Type, Security	Public

Document Title:	Event Discovery by clustering
Agreed date of delivery	
Actual date of delivery	
Editor	TUT: Riitta Kerminen, Jarmo Makkonen, Ari Visa Nokia Research Center: Igor Curcio, Sujeet Mate
Version and	version 2.0
Date Last Change	13.9.2010
File:	

Participants	Name	e-mail
Tampere University of Technology	Riitta Kerminen	Riitta.kerminen@tut.fi
	Jarmo Makkonen	Jarmo.makkonen@tut.fi
	Ari Visa	Ari.visa@tut.fi
Nokia Research Center	Igor Curcio	Igor.curcio@nokia.com
	Sujeet Mate	Sujeet.mate@nokia.com

Table of contents

Table of contents	2
1. Introduction	3
2. Methods for finding events	4
2.1. Limitations that apply	4
2.2. Distance measures	5
2.2.1. GPS Location	5
2.2.2. Record date (Time)	5
2.2.3. Text fields in metadata (Keywords)	5
2.3. Keyword extraction from a (sets of) videos	6
2.4. Missing data compensation	7
2.5. Data-driven method	7
2.5.1. Basic algorithm:	7
2.5.2. Mapping clusters to query	11
2.6. Query driven method	12
3. Test data set	13
4. Tests	15
4.1. Performance measures	15
4.2. Tests	16
4.2.1. Cluster seeds (Time and location)	16
4.2.2. Text clustering	16
4.2.3. Adding videos to cluster seeds based on keywords	18
4.2.4. Adding videos to cluster seeds based on keywords and extra data - Basic algorithm	23
4.2.5. Adding videos to cluster seeds based on weighted majority voting	24
4.2.6. Mapping	25
5. Analysis of the test results	26
6. Conclusions	27
7. References	28
APPENDIX A: Missing Data Compensation Test Report	30
1 Introduction	30
2 Text-to-GPS conversion	30
2.1 Dataset and algorithm	30
2.2.2 LocationText	31
2.3 Google Geocoding ((normally 1 result per word)	32
2.3.1 All text fields (title, description, keywords, location)	32
2.3.2 LocationText	32
2.4 Conclusions	32
3 Text-to-DATE conversion	32
3.1 Description	32
3.2 Results	33
3.3 Conclusions	33
References	33

1. Introduction

This document describes algorithms that can be used for finding events from large real world data sets. The methods are based on clustering in such a way that each cluster represents an event. An event is defined as something that happens in a single place or area, during a time, typically ranging from an hour or two (a concert/football match) to a few days (a festival i.e. Roskilde in Denmark). This definition makes some events problematic, i.e. New Year that takes place almost all over the world, but nearly simultaneously.

We have used YouTube for gathering our test data set. Due to the nature of the data, we have decided to try out several different ways of doing the clustering. (1) can be used independently, but is also a part of (3). (2) is used independently. (4) is not used independently, but as a tool for enhancing (3):

1. The first way (CA1) is to cluster the videos based on their time and GPS metadata. This is quite trivial, because a large number of good clustering methods exist. The problems are related to the data set size and the event definition: In how large an area and at how long a time can an event take place? The clustering algorithm generally has to make an implicit assumption of this.
2. The second intuitive way (CA2) is to use text clustering. We can assume that the metadata of the videos from the same event contains similar text and keywords. This is useful because in the real world data set, only a small portion of the videos contain GPS or record time metadata.
3. Third (CA3), we will combine these clustering methods to hybrid methods that try to exploit the high accuracy of GPS and record time metadata, and compensate missing metadata by using keyword matching.
4. Fourth, we explore whether missing metadata values can be harvested from the metadata text and used in a clustering algorithm.

This document is divided as follows: Chapter two presents our methods for performing the clustering. In the third chapter, our test data set is described. Chapter four presents the tests and their results, while chapters five and six are reserved for test result analysis and conclusions, respectively.

2. Methods for finding events

2.1. *Limitations that apply*

Clustering algorithm performance is a tradeoff between accuracy and computational complexity. Due to the gigantic and ever increasing number of videos in the web, it is impossible to think of clustering a whole data set at once. Becker et al. [1] have considered a problem similar to ours, and have come to a conclusion of which algorithms can and which ones cannot be used.

Suitable:

- Threshold-based methods
- Online/Incremental clustering

Not suitable:

- Hierarchical clustering (Need to calculate distances to all samples over and over again) [17]
 - Hierarchical clustering is ideal for cases where data has a hierarchical inner structure. The algorithm combines always objects with smallest distance from each other. Sub clusters are created and combined to other sub clusters and objects. Therefore distances are calculated repeatedly across the data and clusters. Desired level of clustering is selected from the end results.
- K-means, EM (make assumptions about the number of clusters and their size) [17]
 - K-means algorithm is a greedy iterative algorithm that attempts to minimize sum-of-squares criterion calculated inside each cluster by updating the mean vector until there is no change in the update. Knowledge about the number of clusters is needed before the algorithm is run.
 - Expectation maximization (EM) is another iterative algorithm that is used in search of a best model and attempts to maximize log-likelihood function. Problem with this approach is that it searches the optimal solution, i.e. heavy calculations for iterative algorithm.
- Scalable graph partitioning methods [17]
 - Scalability here refers to the ability of adapt to high dimensionality of the data. Graphs are based on defining nodes and connections between the nodes. The system is uselessly complex for this solution, and we do not have link information between the videos available at this point.

These algorithms are the ones that will be used mainly for clustering algorithms CA1 & CA2, but the same principles are also present in the algorithm CA3.

2.2. Distance measures

In clustering, we need to measure similarity between objects, in our case video metadata. For each component of the metadata, similarity scores can be defined. The following measures are applied in our methods.

2.2.1. GPS Location

Distance between two locations is calculated from GPS latitude and longitude coordinates with the Haversine -formula. It uses the angle coordinates to calculate distance between two locations along approximated earth surface. The formula is generally used and several web pages offer more information about it, this definition was in [2]. The definition for the distance between two GPS coordinate points is:

$$\begin{aligned}
 R &= \text{Earth's radius (mean radius = 6,371km)} \\
 \Delta lat &= lat_2 - lat_1 \\
 \Delta long &= long_2 - long_1 \\
 a &= \sin^2(\Delta lat/2) + \cos(lat_1) \cos(lat_2) \sin^2(\Delta long/2) \\
 c &= 2 \operatorname{atan2}(\sqrt{a}, \sqrt{1-a}) \\
 d &= R c
 \end{aligned}$$

(Note that angles need to be in radians to pass to trigonometric functions).

Latitude and longitude are marked as *lat* and *long* for each coordinate point.

2.2.2. Record date (Time)

The time field in the video metadata has an accuracy of one day. This is because of the limitations of YouTube metadata. Usually, in the official format, the date is given as yyyy-mm-dd. In a preprocessing stage, each date was converted to a serial date number. The serial date number of 1 corresponds to Jan-1-0000 and is meant only to be used as a reference point. Accordingly, the distance between two dates is a simple subtraction between these values. That also means that preceding days have the distance of one.

2.2.3. Text fields in metadata (Keywords)

Distance between keywords is actually the distance between keyword lists. The test data is mainly in English, so similarity measures (e.g. Levenshtein [15]) between actual words are not needed. Simply, we divide the text fields into words, remove any stop words (extremely common words, [16]) that contain little or no information, and compare the words with a scale {matches} / {does not match}. The distance of two keyword lists can be calculated several ways. If two word lists are near same lengths, for example in two video titles, we can use the normalized word list distance. In this case it is calculated as:

$$\text{Distance}(wl1, wl2) = 1 - (\text{number of same words in } wl1 \text{ and } wl2) / (\text{average length}(wl1, wl2))$$

This way the distance is 0 if the lists are the same and 1 if they have no common words. The drawback of this approach is that the word lists need to have nearly the same lengths to be useful.

Another way to calculate the distance between keyword lists is to take the percentage videos that have similar words in the keyword fields. This approach needs some normalization, for example deciding beforehand the length of keyword lists.

2.3. *Keyword extraction from a (sets of) videos*

In the algorithm CA3, keyword extraction is needed. This means that we need to divide the text fields in a set of video metadata into words, and select the most representative of them as keywords for that set of videos. This can be done in many ways.

Generally, the relative importance measure Term Frequency - Inverse Document Frequency (TF-IDF) has been found useful in selecting keywords [1,4,10]. The idea of the approach is that words, that are common within a document, and relatively uncommon in the global set of documents, make good keywords. However, we find that it is not suitable for our needs because for that to be effective, we need the appearance frequencies of all possible words in all documents in a video database. For a small set, this is easy to gain. However, in a large environment, like YouTube, it is impossible. Moreover, with a dataset of size that is large enough, the appearance frequency of a word is close to any word in a language. Thus, we argue that it is enough to remove the known words of low relative importance, stop words. In addition, in our test case, the small database size means that the type of documents is biased towards the type of events we have in the database, and thus TF-IDF might not suggest good keywords.

We denote a set of videos, say, the videos in a cluster, with $V = \{v1, v2, v3...\}$. We define a parameter $N_keywords$ to represent the maximum number of keywords to be selected from each video set.

All the words (stop words removed) in all videos of the set V are calculated. Then, the words that appear the most times are selected to be the keywords, until $N_keywords$ is reached or all words are assigned as keywords. There is no minimum number of appearances that a word should have to be selected.

We have implemented also other versions of the algorithm. Instead of just plain number of appearances, the selection criteria of the keywords could include that the word should appear in all videos of the set. This would restrict the amount of candidates severely, though. Another method that has been considered is to give the words a score that would differ from the number of appearances. The score could be calculated by giving weight to the word appearances depending on which text field they are in. For example, we could suppose that the words that appear in the "keywords" text field are of more importance than those appearing in the

“description” field. Thus, we could give a higher score to words from “keywords” field.

2.4. *Missing data compensation*

Only some videos in our data set contain GPS and date values as metadata. On the other hand, they are crucial for the performance of the clustering. Thus, we studied whether these values can be derived from the text fields. The report of the study is included as Appendix A.

The missing data compensation is tested with the algorithm CA3.

The use of missing data compensation in creating new clusters requires some modifications to the clustering algorithms. The clusters have to contain a probability measure and that has to be carried all the way in the process. Moreover, the complexity of the algorithms rises many steps up.

In the time frame of this project, we did not have time to develop a working version of this kind of approach.

2.5. *Data-driven method*

Here, by data-driven, we mean that the clustering is controlled wholly by the data, and happens regardless of any user interaction by e.g. a query (“I want to have all videos from Tammerfest 2007”). The database is clustered into events and structured in some clever way that a cluster that matches any query can be returned for the calling application. The following are the three main methods for building these clusters.

2.5.1. Basic algorithm:

The clustering is done with incremental clustering, so that distances for the new video are calculated between cluster centers and the video. The video is included in the first cluster which satisfies threshold values for each distance.

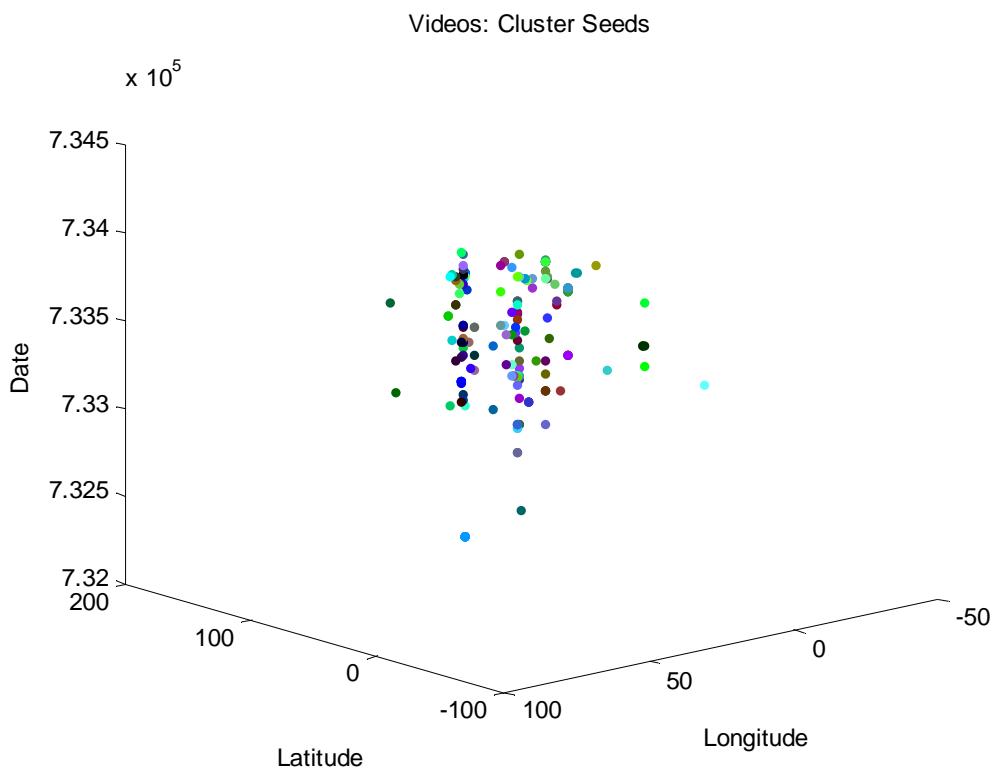
Pseudo code for the algorithm:

- Get a video metadata
- Compare the data to previous clusters
- If distance between cluster center and metadata small enough (Threshold),
 - Add video to the cluster
 - Adjust cluster values
- Else
 - Test next clusters
- All clusters been tested and no match
 - Create a new cluster
- Return to the beginning of the algorithm

One common threshold is used for all videos.

Clustering by time and location (CA1)

The first algorithm is a straightforward clustering based on video location (GPS) and a timestamp. Figure 1 presents a clustering result as a 3-dimensional scatter. When this information is available, our task of finding event clusters is trivial. Several clustering algorithms exist in literature [11, 17]. The biggest practical problems to consider are the ones mentioned in 2.1, and the problem of how to efficiently store the cluster and video data to enable light mapping and new video addition algorithms. This, however, is out of the scope of our research.



1. Figure: Clusters generated by CA1. Each dot represents one video in a space formed by time and 2D-location (GPS). Same colour means that the videos are in the same cluster.

Clustering by text (CA2)

There are several algorithms to cluster text. Depending on text file, for example long documents vs. text messages, different approaches work best. Videos title and keyword fields contain very precise description of the events. On the other hand, description text contains more complete sentences and therefore, from the event clustering point of view, useless words. Therefore, current results of pure text clustering are calculated from title fields.

Keyword extraction is an important part of the text clustering. The issue has been discussed in chapter 2.3. Basic clustering process is shown in chapter 2.5.1. Text clustering is not very accurate method. Words people use to describe or name

events are not precise and often mean more than one thing. This is especially evident in band names or names of the songs. For example, YouTube search for song 'Speed' from band 'Analog Fish' results in one performance by the band of the song and other videos containing instructions how to draw fish fast or have the music as a soundtrack. Same event is also often written down with different words. Spelling errors add one more layer to things to overcome. Some of the text clustering could be made more accurate by adjusting threshold and keyword selection values. Also clustering algorithm and distance measure could be changed. Also for future testing using weighted keyword selection could offer more possibilities [3].

Hybrid clustering CA3

The algorithm starts by clustering videos that contain both time and GPS metadata. This is done with algorithm CA1. The resulting clusters of this phase are called the cluster seeds. No further clusters are formed after this phase. The principle of the algorithm is shown in figure 2.

The second phase is to add more videos to the cluster seeds. These are the videos that lack the GPS or time metadata, or both. For each cluster seed, representative keywords are selected. This is done by using a keyword extraction algorithm. Different algorithms are described in 2.3. Moreover, the cluster seeds are given a cluster centre that is calculated as the mean of the time and GPS values of the videos within the cluster seed.

The actual video addition is done by browsing through the remaining videos. There are two ways of doing this. The first way is a heuristic-based approach. A similarity score is calculated for each video against each cluster. All videos that fit to the existing cluster seeds are added to them. The similarity is calculated as follows:

$$\text{Similarity}(\text{video}, \text{clusterSeed}) = \text{matchingKeywords}(\text{video}, \text{clusterSeed}) + \text{weight1} * \text{matchingLocation/Time}(\text{video}, \text{clusterSeed}) + \text{weight2} * \text{matchingExtraLocation/Time}(\text{video}, \text{clusterSeed})$$

The functions "*matchingLocation/Time*" mean that either the time or location values are close to each other. The functions "*matchingExtraLocation/Time*" are the same for the values harvested by the missing data compensation.

The weights that are used are given as parameters. Moreover, the similarity score has to be over a certain threshold to be selected into a cluster. These weights and thresholds are one of the main things to be studied during the tests.

Another way of adding videos to cluster seeds is to use a weighted majority voting system. The principle of the algorithm is shown in figure 3. The system works in a similar fashion as the heuristic-based one, but it allows more flexibility and thus can be trained for optimal performance. For every piece of metadata we have,

there is an expert. Every expert gives an opinion on how probable it is for a video to be a part of each cluster based on one type of metadata. The outputs of all experts are multiplied with some weights, and the result is connected as a sum. If the sum is greater than a threshold, then the video is accepted, otherwise not. The weights and the threshold are given as parameters, and they can be optimized with some learning method.

The videos that did not fit to any of the clusters could then be used for creating new clusters. This could be done by compensating the lack of GPS or time metadata by using either keywords or values harvested by the missing value compensation.

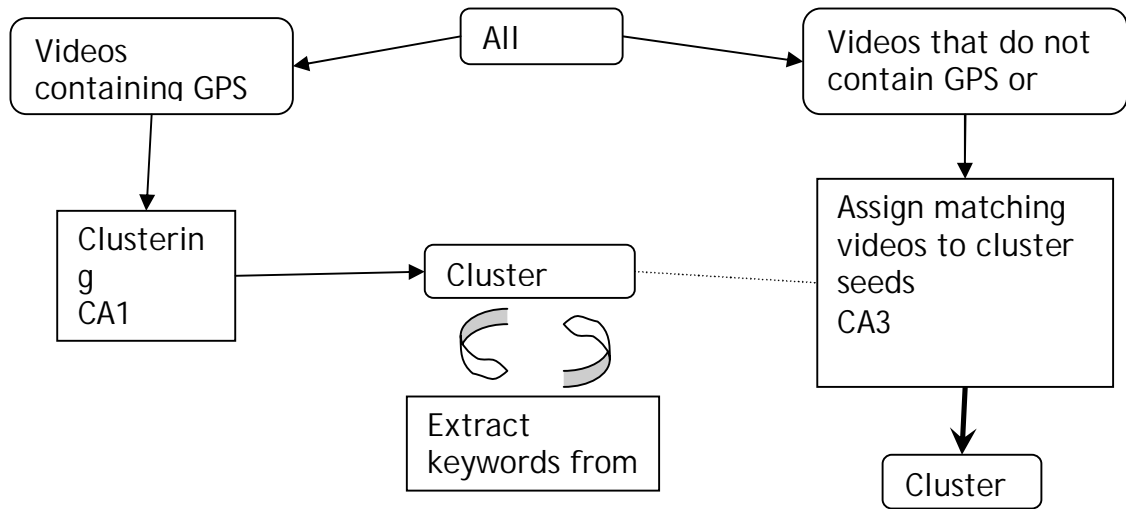
During this project, we have not had time to find a working algorithm for creating new clusters.

This method has been discarded because it was too complex to implement, and did not show enough potential to be useful.

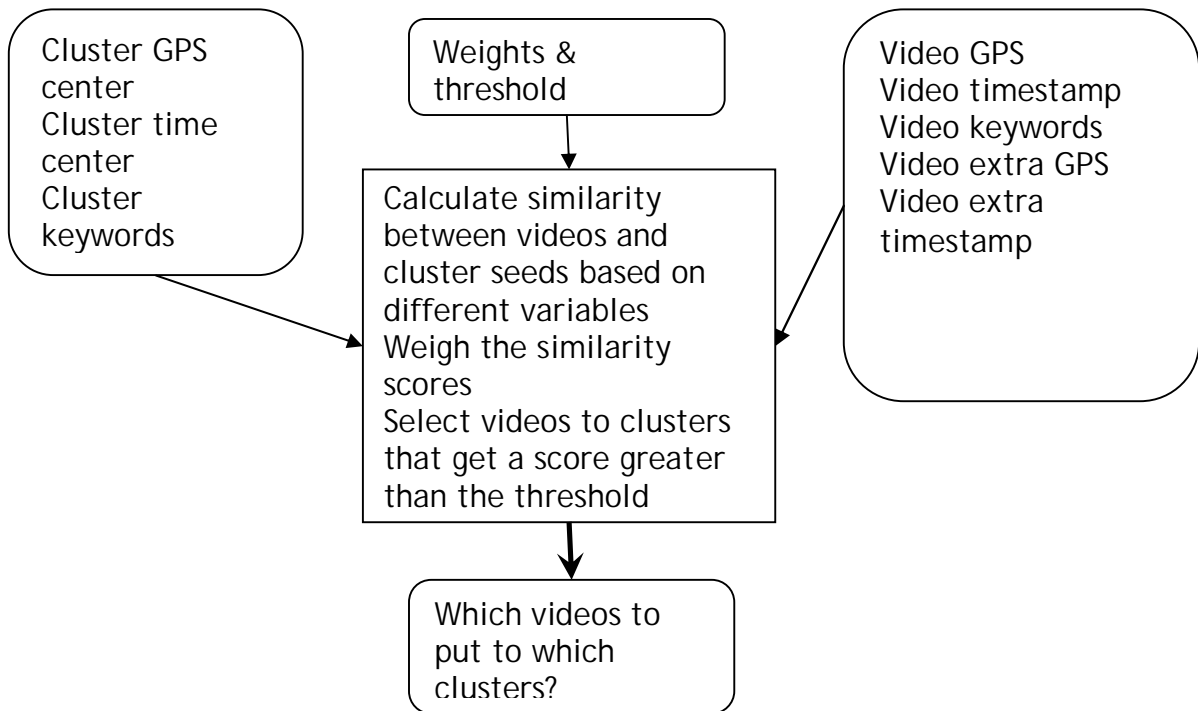
This clustering method is like the first hybrid algorithm, but works in the opposite way. Instead of time and place clustering, we consider the text clustering as the starting point. Clusters will contain videos that are similar in terms of their text content. However, they may contain videos that do not belong to the same event in terms of time and GPS information.

Thus, the algorithm does a second clustering step. It breaks the clusters up into groups, first based on their time values, and then based on their GPS -values. Each of these groups will be considered a small sub-cluster. Those videos that do not have time or GPS-information will be inserted in the sub-clusters based on keyword matching.

In the third step, the algorithm tries to combine the sub-clusters based on the time and space-values. The clusters will not be broken down at this point.



2. Figure: Algorithm CA1 and CA3 operation



3. Figure: Principle for CA3 operation

2.5.2. Mapping clusters to query

User queries are mapped to the clusters based on the distance between the cluster and the user query. The distance is calculated by a similarity metric. Similarity

based on time and location is straightforward to calculate. Distance measures have been discussed in chapter [2.2](#).

The mapping results are presented as a list of videos in the closest cluster. Each cluster has a distance to the query. Again there are threshold values that decide if the cluster should be accepted as close to query. If all of the query parameters are fulfilled, the system returns only these cases; otherwise the system returns all clusters that have even one query parameter fulfilling the query. Mapping result listings should be adjusted to fit an application. Negative side for this approach is that the number of clusters is always increasing and therefore distances calculated per query increase processing time for queries.

2.6. Query driven method

Query-driven event search means that there are no or only partial pre-calculated event clusters in the database. This leads inevitably to longer response time for the user. Moreover, more calculations have to be done than in a database-driven case because each query launches an event search process.

A naive way of doing the search would be to calculate the distance of the query parameters to all videos in the database. In the case of billions of videos, this is clearly not the way to do it. The database should be organized cleverly to enable narrowing down the search to only those videos that are from the same time and place as the query parameters suggest.

This, however, leads us to a situation where preprocessing in the database is needed. Also, if the query contains only text, clever database organization is harder to achieve. Based on this knowledge, there are at least two promising approaches that could be used.

The first one is to use a Self-Organizing Map (SOM, [12]) that organizes similar videos close to one another. The downfall of this approach is that the resulting SOM is always a black box, and that it requires a large annotated training data set.

The second option is to perform the clustering as in the database-driven case. Then we can order the database based on the cluster centre coordinates and time stamps. This means that the clusters that contain neither one of these cannot be used. We then map the used query to the database, and select a small number of closest clusters. We can then use the videos in these clusters as the promising data set for the query-driven search. This can be done simply using the naïve search. The resulting video set contains those videos from the mapped clusters that match the query.

We have implemented and tested this algorithm, but no thorough test results are provided in this document because of the simple nature of the algorithm.

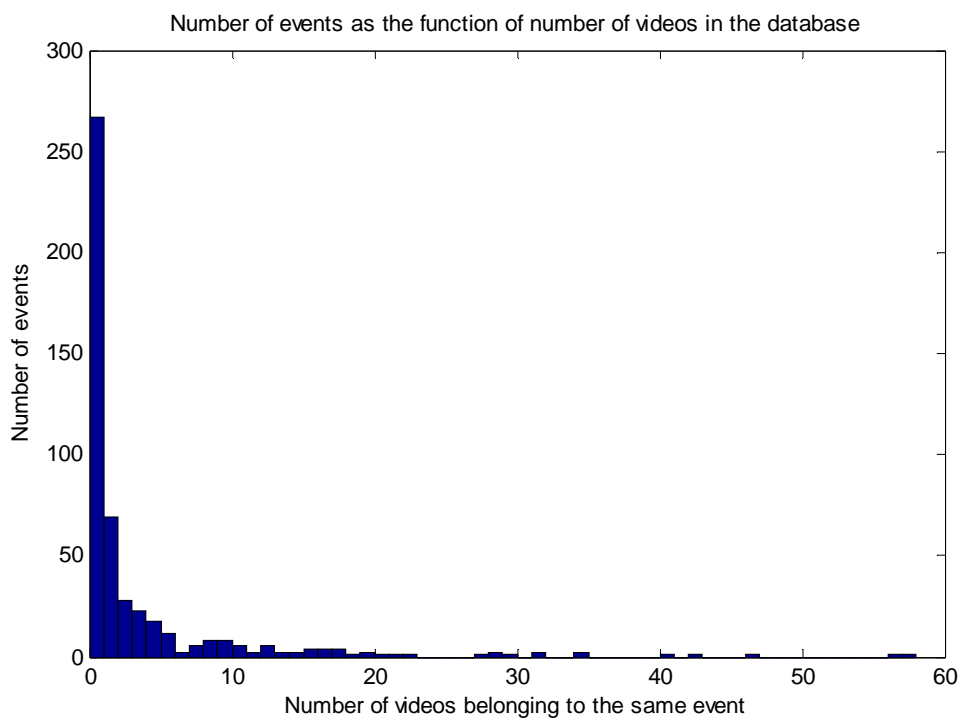
3. Test data set

We have gathered test data from YouTube by using a PHP script that fetches the video metadata in XML format. The data was then stored in a MySQL database. The test data was selected manually by watching the videos. These videos have been annotated and given an event identifier number that acts as the ground truth for testing. We have also gathered automatically a large database that has not been annotated. This database is combined with the annotated base to represent noise.

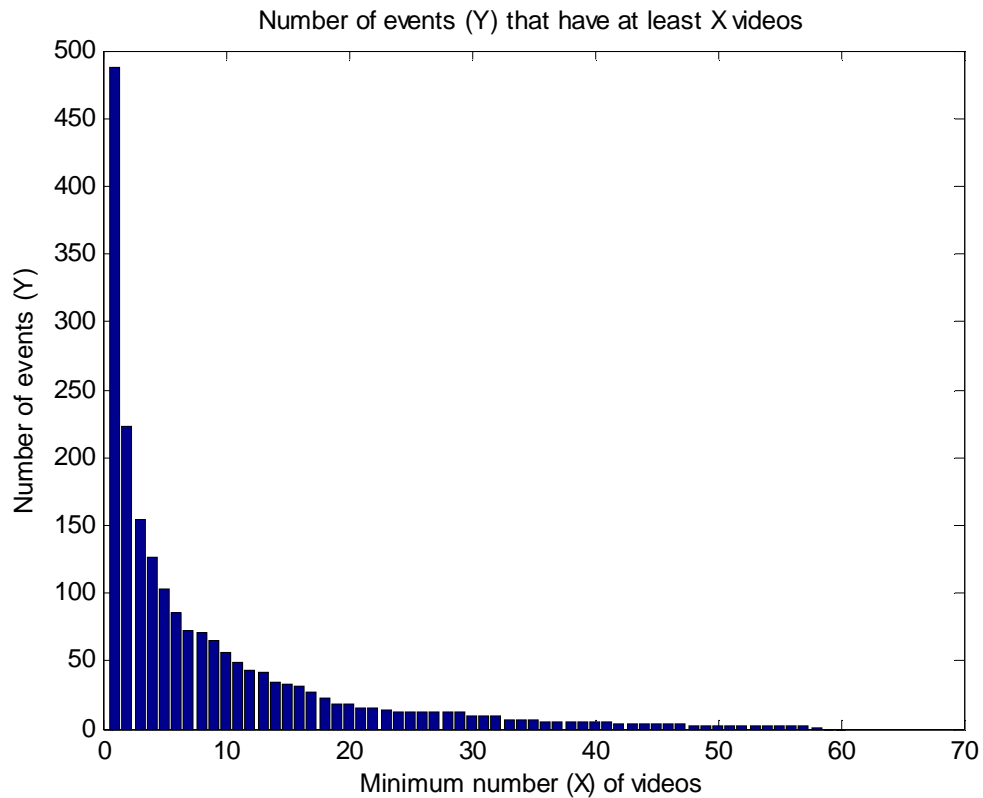
The basic statistics of the annotated database are as follows:

- 488 annotated events in total are in the database
- The database contains 2021 videos
 - 265 events contain only one video
 - 103 events contain 5 or more videos
 - 107 events have a video that contains both time and GPS information

Figures 4 and 5 represent these statistics in detail. Table 1 gives an overview on the metadata structure.



4. Figure: Number of events that have certain number of videos



5. Figure: Number of events that have a minimum number of videos

id	yt_id	title	description	category	keywords	GPS	location	recorded
15	xyz123456	Liverpool-Sunde	Liverpool-S	Sports	Anfield, L	53.430721	N/A	2008-02-02
37	zyx987654	ManU v Liverpool	Dossena scd	People	MUFC, LFC	53.462772	N/A	2009-03-14
57	someytID	FC Barcelona vs R	Visca Barca	Sports	FC, Barcelo	N/A	Barcelona, Sp	2007-12-23
77	TUT99887	icehockey, Tappa	The better t	Sports	icehockey,	N/A	Tampere	2007-09-18
205	kakfka233	Iron Maiden perf	Iron maider	Music	tampere, i	N/A	tampere rati	2008-07-19
259	fjafSF2982	Mikontalo Lights	Tetris on a b	Tech	window, n	61.446544	N/A	2007-12-02
350	fldsajfl298	Oktoberfest Mur	Oktoberfes	Travel	Oktoberfe	N/A	N/A	
1036	SGNTUT32	Tammerfest 2009	Tampere su	Music	kolmas, na	N/A	N/A	

1. Table: Basic structure of database and sample data (shown data values are not actual YouTube data)

4. Tests

4.1. Performance measures

Measuring the goodness of clustering is complicated, because unlike in a classification problem, we do not have a set of predefined classes with labels. We know only, how the videos are divided into clusters, but it is unknown if a video is in the “correct” cluster. For example, the number of created clusters can be different from the number of actual events available in the data set.

To overcome this difficulty, there exist several statistical measures that represent the goodness of the clustering. For example, cluster purity defines how much the clusters consist of only one “class” (event annotation in the data set). The measures that we use (described in detail in [13]) are:

- Entropy score: A measure for the amount of information. The more homogenous the clusters are, the less information they contain. (0 is best, i.e. clusters contain no information i.e. all videos from same event)
- Purity score[14]: 1 is best (clusters consist of only one event)
- Average purity (The sum of all cluster purity scores divided by the number of clusters)
- Percentage of events found (of all possible)
- Different versions of Rand index (A common clustering performance measure) [13]:
 - RandIndex: Hubert & Arabie adjusted Rand index
 - AR: Adjusted rand index
 - RI: Unadjusted rand index
 - MI: Mirkin’s index (Measures dissimilarity of two partitions)
 - HI: Hubert’s index
- Values are mainly affected by the amount of available data

The following metrics are calculated for the CA3 video matching phase. They are based on the assumption that the correct cluster where a video should be inserted is known. In our case, we take that information from the cluster seeds.

False means “wrong decision”, true means “correct decision”. Positive means “video is inserted into a cluster seed”, negative means “video is not inserted in any cluster seed”. I.e. False negatives means the number of those videos that should have been inserted into some cluster, but were not inserted. Sensitivity describes the amount of videos that were correctly inserted in the clusters, out of all those that should have been inserted. Specificity describes the amount of videos that were correctly left out of the clusters, out of all those that should have been left out.

- Total Hit Rate= $(\text{True Positive} + \text{True Negative}) / \text{All results} (\text{True Positive} + \text{True Negatives} + \text{False Positives} + \text{False Negatives})$

- Positive Hit Rate= True Positives/ (True Positives + False Positives)
- Negative Hit Rate= True Negatives/(True Negatives + False Negatives)
- Sensitivity= True Positives/ (True Positives + False Negatives)
- Specificity= True Negatives/ (True Negatives +False Positives)

4.2. Tests

4.2.1. Cluster seeds (Time and location)

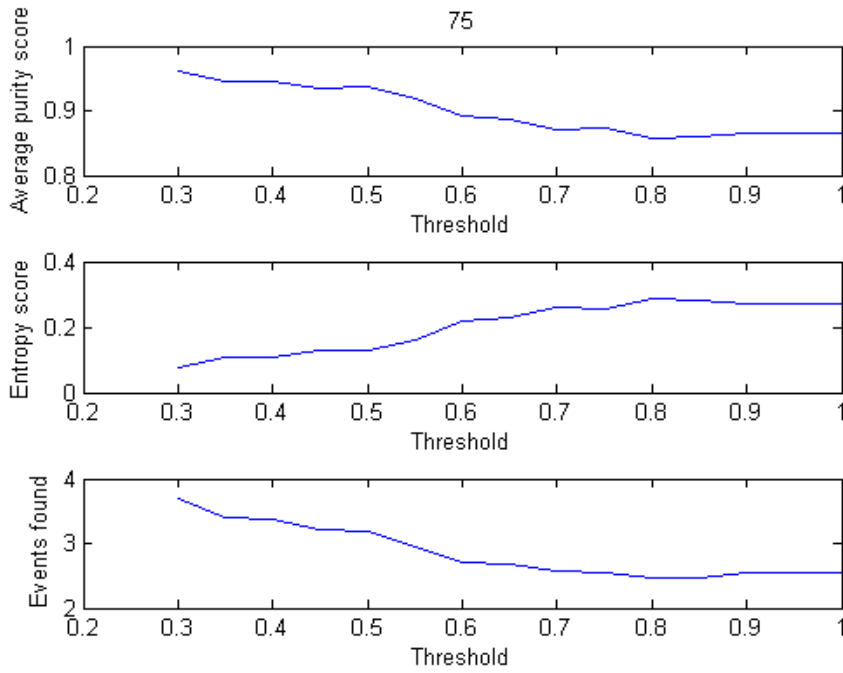
The time and location clustering has proven to be powerful in separating the test database into events. However, testing this clustering with such low density data is not very informative. Separating a relatively small number of GPS locations with time stamps into events is quite trivial, and normally we get a 100% accuracy result. However, sometimes the videos from the same event are separated into two events, due to the threshold values that are used for determining “what is an event”.

4.2.2. Text clustering

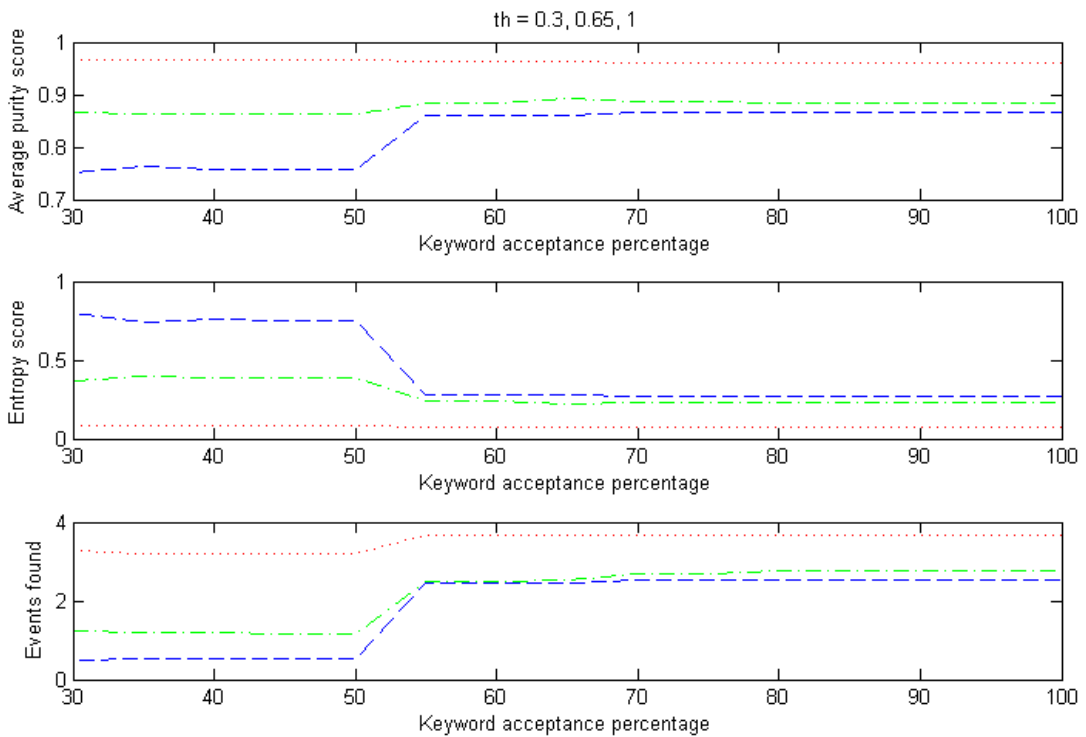
Text clustering parameters were tested with 1000 first videos in the database. The results were calculated and here are some of the figures showing results. Percentage of the events found and purity score are the most telling measures to judge the results. Therefore, figure 6 shows purity scores, entropy and normalized number of events found against keyword thresholds. Keyword threshold is used while counting distance between video and cluster center. If threshold is 0, distance between wordlists needs to be 0, i.e. wordlists are same. Selecting keywords in text clustering is done by calculating appearance percentage for each word for all videos in a cluster. Keyword acceptance percentage in the figure 6 is 75, i.e. the word needs to be in 75% percent of the videos in the cluster to be accepted as a keyword.

In figure 7 average purity score, entropy score and number of events found are plotted against keyword acceptance percentage results. In the subplots distance measure threshold has three values; on red: 0.3, green: 0.65 and blue: 1.

Generally can be said about text clustering parameters is that with small threshold values, videos needs to be very near to the cluster values to be accepted to the cluster. Therefore clusters are more pure and contain fewer videos from several clusters. On the other hand, number of events stays too high, because some videos that should have been accepted to other clusters, were not, and new clusters were created. Keyword acceptance percentage has similar effects. When keyword acceptance percentage rises average purity rises and number of clusters rises. These are general ideas for the parameter. Actual parameter values are somewhat data dependant.



6. Figure: Average purity score, entropy score and number of events found in text clustering.



7. Figure: Average purity score, entropy score and number of events found in text clustering

4.2.3. Adding videos to cluster seeds based on keywords

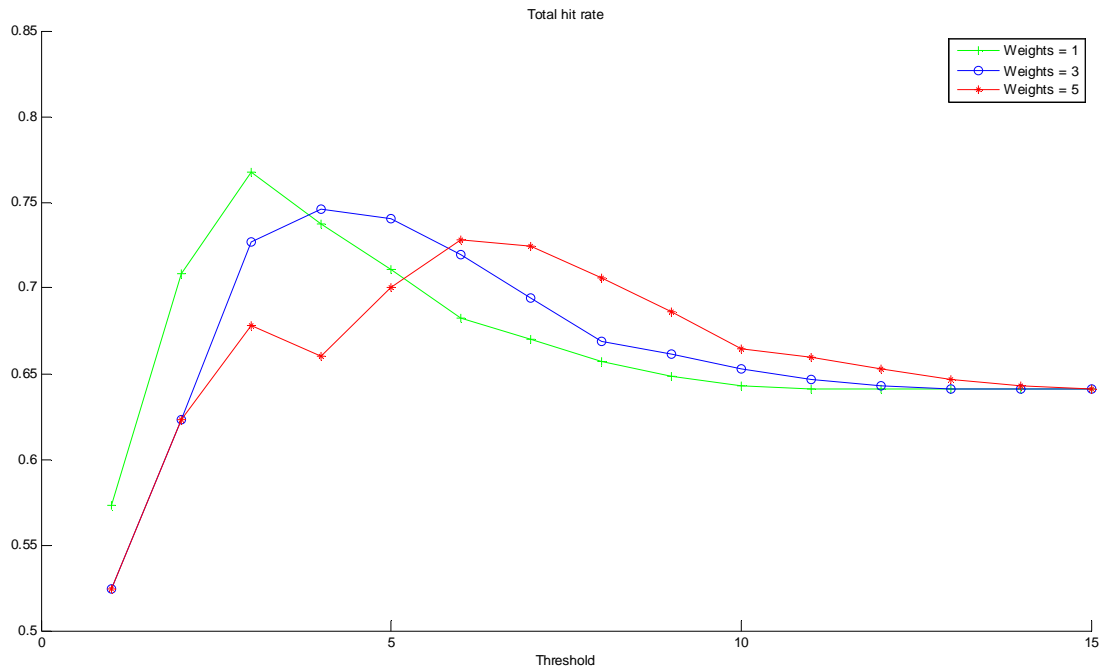
The problem is to decide which of the unclassified videos should go to which cluster seed, if any. In the performance measurement, we state that a video should be inserted to any cluster seed that has the video's true event number as the dominant class. We define the dominant class as the event number that appears in most videos of the cluster seed. Note, that a cluster seed can contain several dominant classes, i.e. in a situation where the cluster seed has 2 videos, both of different events. This is, of course, an error at system level, but should not affect the performance measurement of this phase.

We have run the algorithm for the whole annotated data set with several parameter setups. The following parameters have been used (375 combinations):

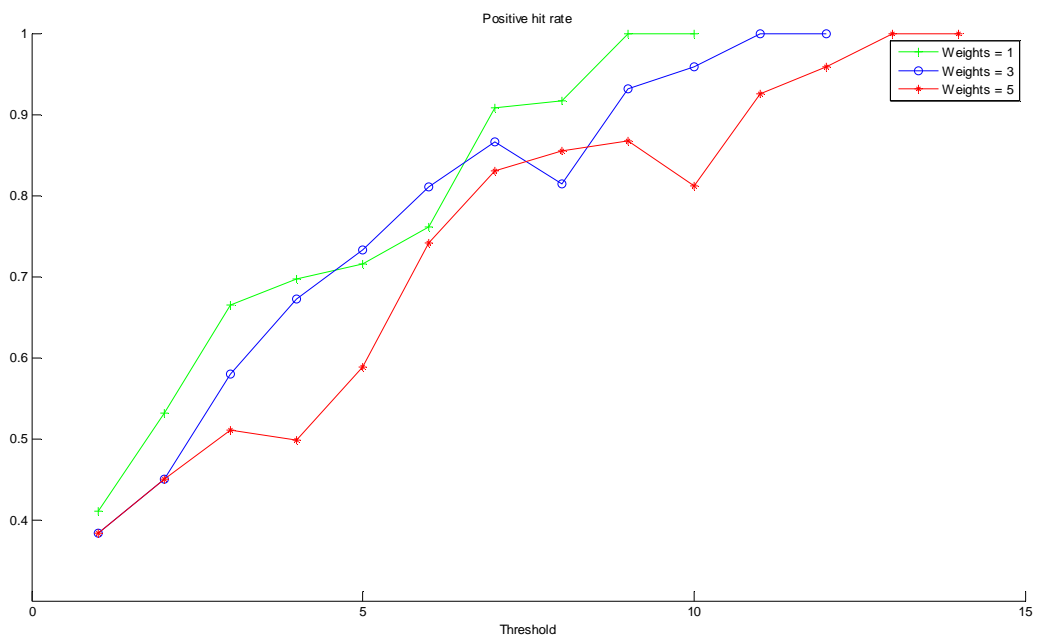
- GPS is close-score weight (How much emphasis do we put on GPS location)
 - 1, 2, 3, 4, 5
- Time is close-score weight (How much emphasis do we put on time stamp)
 - 1, 2, 3, 4, 5
- Score threshold (How likely are we to accept a video to a cluster seed)
 - 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

The test results are presented in figures 8-15. Figure 8 presents the total hitrate for three different parameter combinations. Green line is a case where both GPS and time weights are 1. Similarly, in the blue line case these parameters are 3, and in the red line case, 5. The same colour notation and cases are used in the figures 9-15.

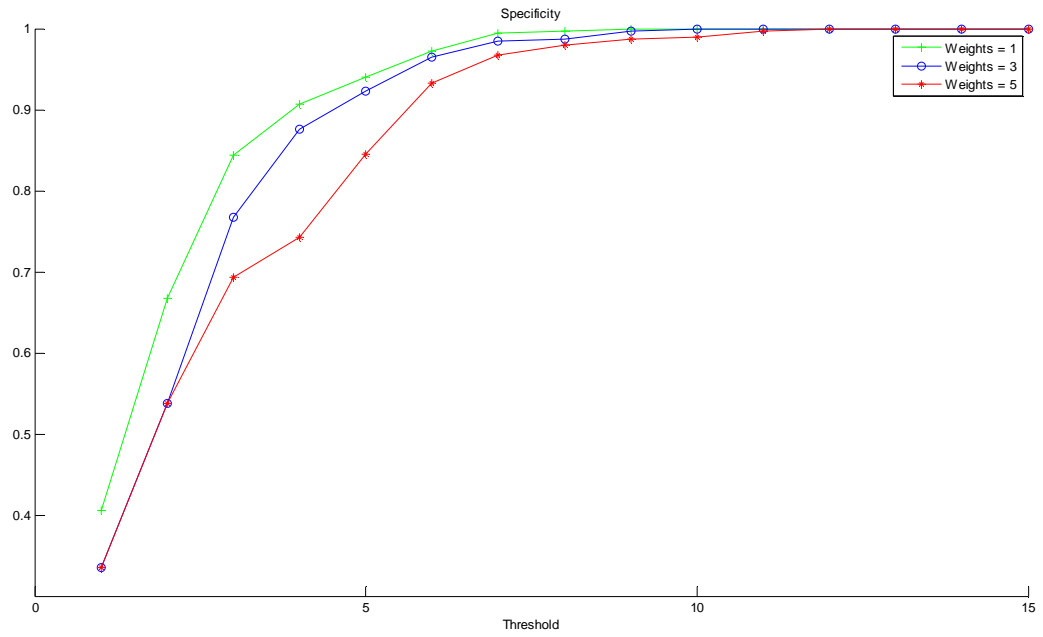
From the total hitrate curves, it can be seen that there exists a global maximum value that appears to be approximately at a threshold of the weight added by 2.



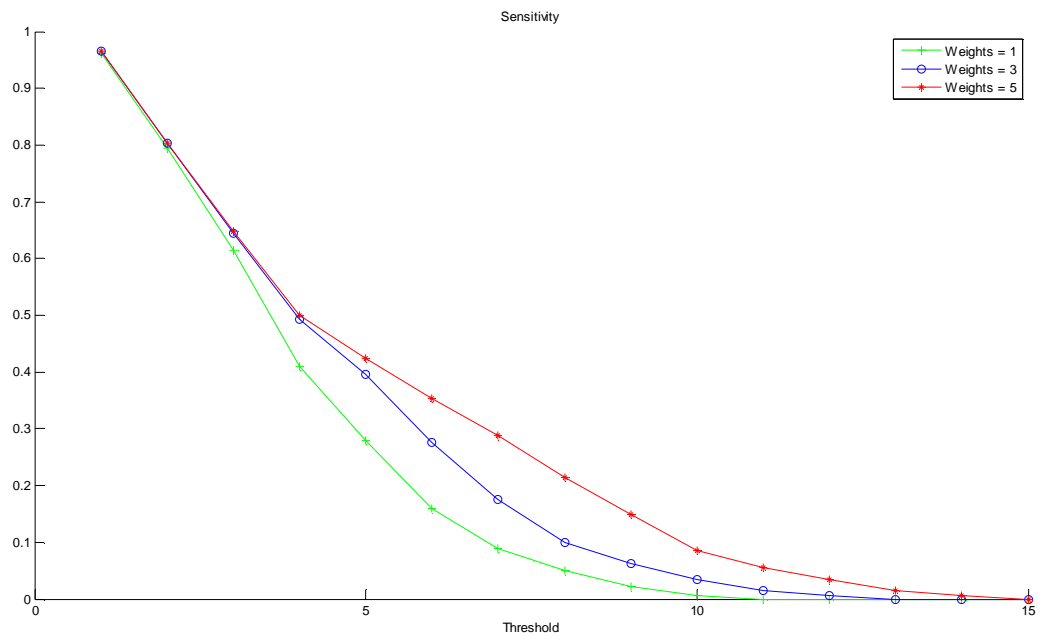
8. Figure: Total hitrate for CA3



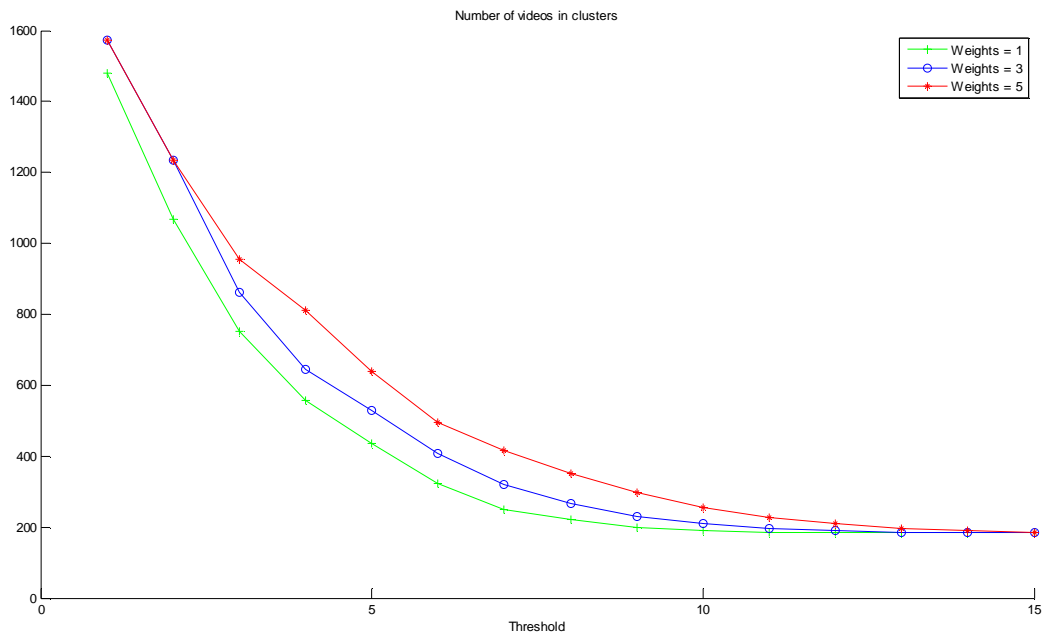
9. Figure: Positive hitrate for CA3



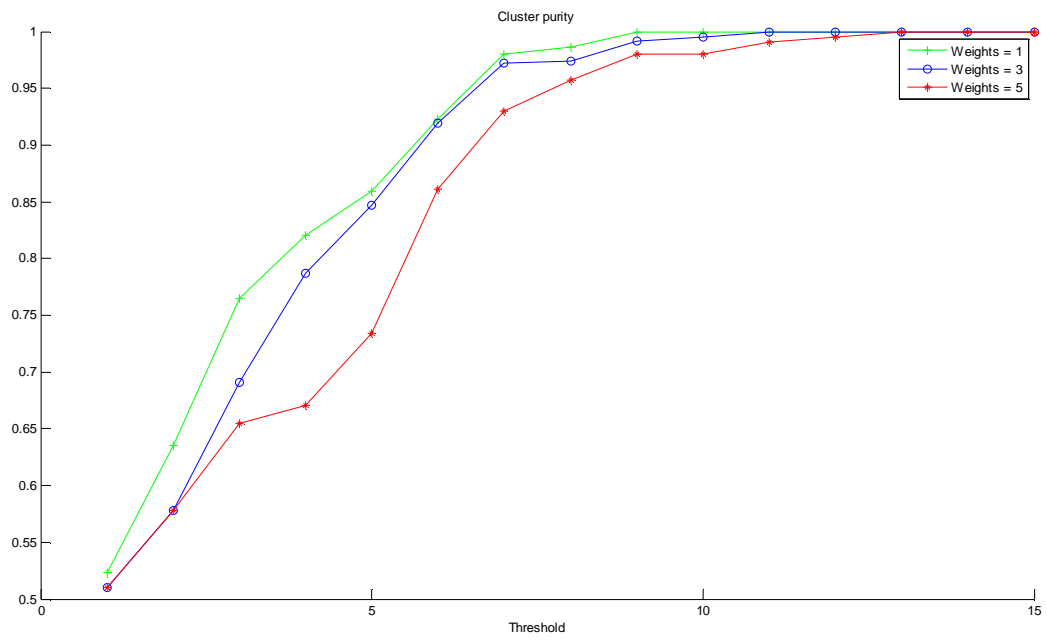
10. Figure: Specificity for CA3



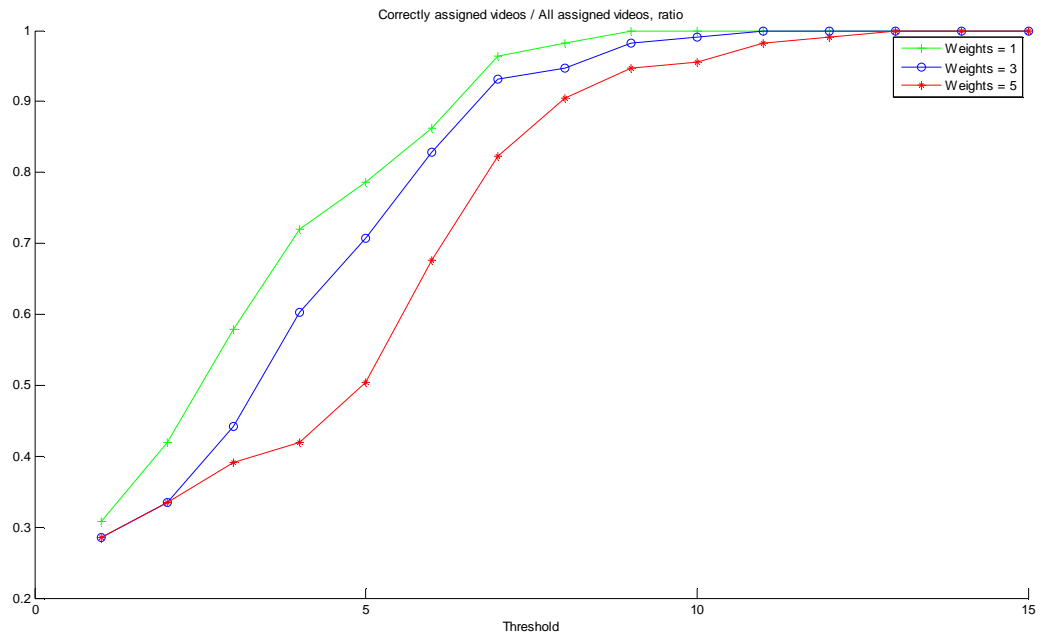
11. Figure: Sensitivity for CA3



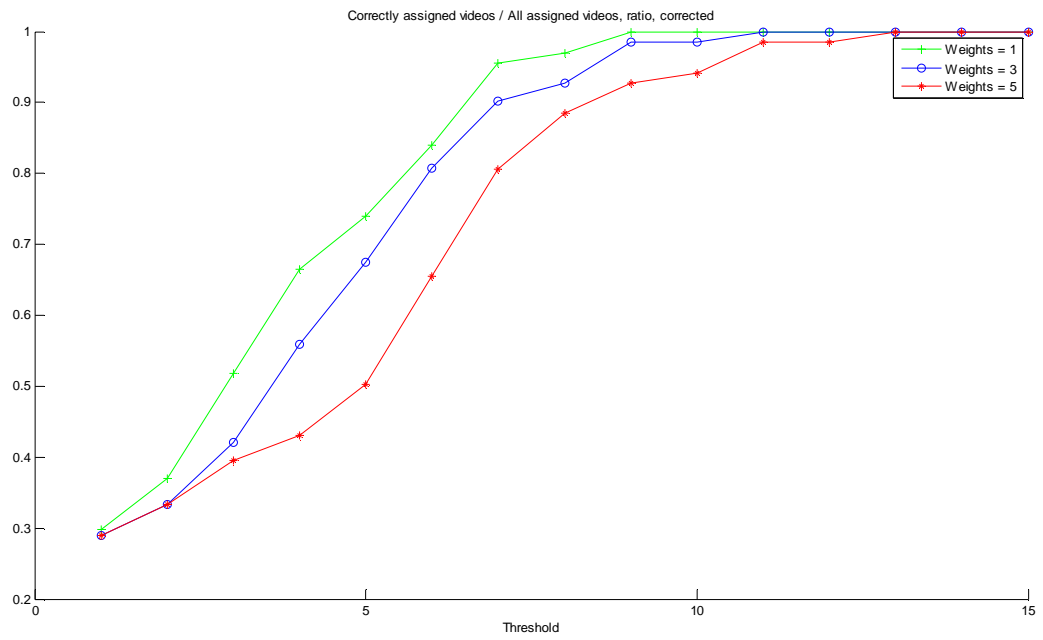
12. Figure: Number of videos clustered in CA3



13. Figure: Cluster purity



14. Figure: Correct videos / all videos ratio, CA3



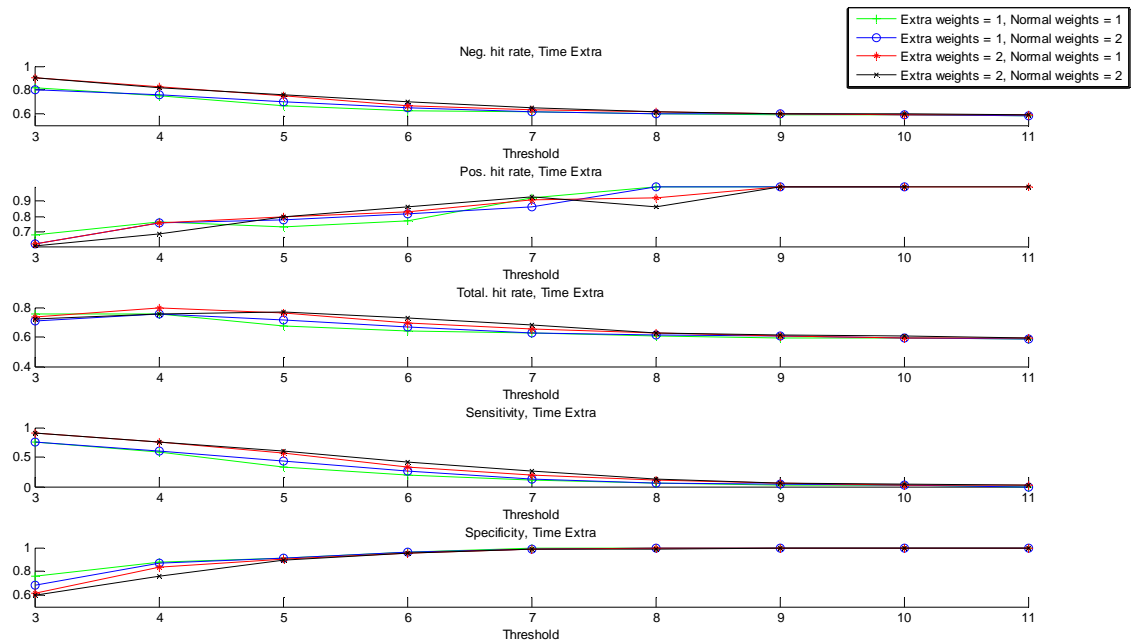
15. Figure: Correct videos / all videos, CA3 (corrected)

4.2.4. Adding videos to cluster seeds based on keywords and extra data - Basic algorithm

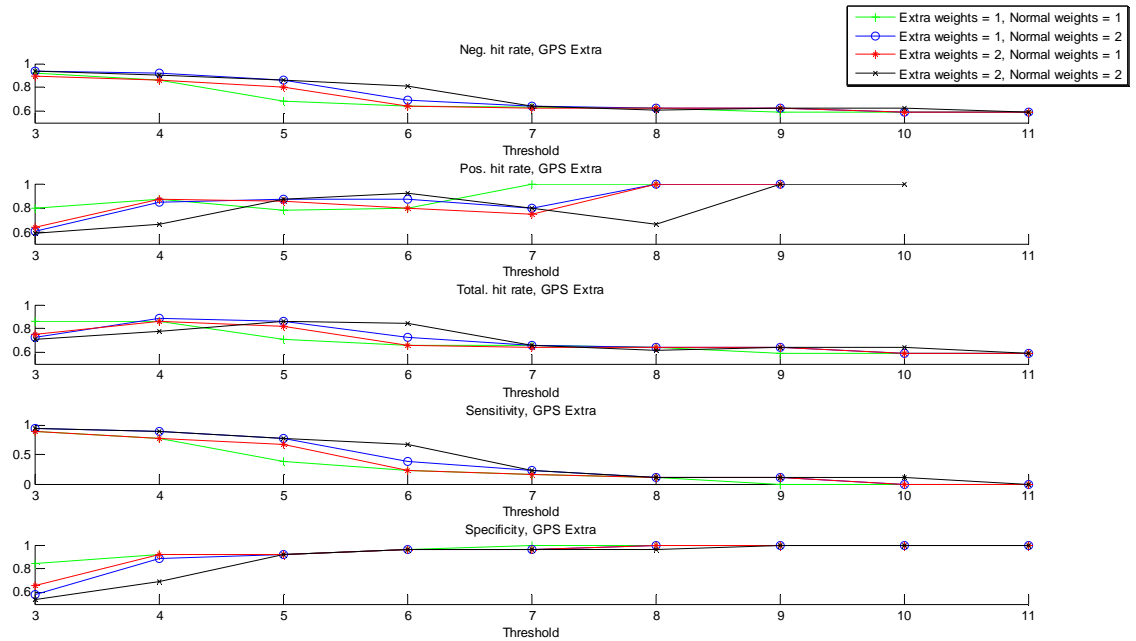
Additionally, we have tested how the use of extra data (GPS and time) can help in this problem. We have run similar tests with the following parameter setups:

- GPS is close-score weight (How much emphasis do we put on GPS location)
- Time is close-score weight (How much emphasis do we put on time stamp)
 - Same value for both: {1,1}, {2,2}, {3,3}
- Score threshold (How likely are we to accept a video to a cluster seed)
 - 3, 4, 5, 6, 7, 8, 9, 10, 11
- Extra GPS is close-score weight
 - 1, 2
- Extra Time is close-score weight
 - 1, 2

Test results are presented in figures 16 and 17.



16. Figure: Performance using extra time data



17. Figure: Performance using extra GPS data

4.2.5. Adding videos to cluster seeds based on weighted majority voting

Here we will have the results for the weighted majority voting. Because of problems in the implementation and the time required to run the optimization for the algorithm, the work has not yet finished.

- The tests were done by dividing the videos that are not included in the cluster seeds into two sets: a training set (every third sample) and a test set (rest of the samples).
- The Optimization Toolbox of Matlab was used for finding the parameter in training.
- The achieved parameters were used and the algorithm was performed on the test set. The key performance measures are shown in table 2.
- The parameter set that was gained by the training phase was {GPSWeight, TimeWeight, GPSExtraWeight, TimeExtraWeight, Threshold, KeywordWeight} = {3.0, 4.3, 2.0, 4.2, 4.5, 1}

Number of videos inserted in clusters	206
Number of videos discarded	1017
Number of correctly/falsey inserted videos	152/54
Number of correctly/falsey discarded videos	742/275
Positive hit rate	73.8 %
Negative hit rate	73.0%
Total hit rate	73.1 %

Sensitivity	35.6 %
Specificity	93.2 %
Cluster purity	87.8 %

2. Table: Performance measures for weighted majority voting

4.2.6. Mapping

Each annotation contains name of the event, location and time. This approximates normal keyword query quite well. For each video, we ran the query with these parameters and checked if the video was included in the found clusters. The test was affected both by mapping, clustering, and annotation. Selecting clusters and querying with their information tests only mapping. Distance between these two is only affected by clustering and annotation. Unfortunately the latter test has unnatural accuracy as query parameters.

The results in table 3 show that mapping works, if the query is done accurately enough. CA3 algorithm kept only the keywords from the original cluster seeds so the annotation did not match all correct clusters. This could be remedied by selecting keywords also from the new videos that were added to the cluster. Also taking into account human factor in the querying might affect the results. Annotations are quite precise search criteria. If we have location and time, CA3 - gives always correct results. Depending on threshold for the distance and time we also might get some extra data.

Clustering, Query, Number of objects	Text, Annotation, 2021 videos	Text, Cluster, 457 clusters	CA3, Annotation 569 videos	CA3, Cluster, 127 clusters
None	0.0094	0	0.2232	0
Correct	0.6126	1	0.7030	127
Incorrect	0.3780	0	0.0738	0

3. Table Mapping test results

5. Analysis of the test results

CA1 seems to work really well as expected. The only problem is that too many clusters are formed, i.e. some events are split to several clusters. This might be because of threshold values.

CA2 yields modest test results, and is not useful by itself. This was known from the beginning. The way how people write descriptions to video metadata does not differ much between the types of content. If text clustering parameters were adjustable during the process, the result might be better. For example, in a case where video title contains information on an artist, location, and the name of the tour, the location information gets too little attention and videos from the same tour but several locations end up in the same cluster. A similar case could be a concert with several artists with different songs and the connecting information is only a couple of words.

CA3 performance depends heavily on the parameter values. By looking at figures 8-15, we can make some notes on how the relationship between the parameters and performance works.

The total hit rate curve is the only one that tells us directly something about the overall system performance. The rest of the measures are tradeoff values. Depending on the weights used, the optimal threshold level seems to be a little over the weight value, 3 to 6. This tells us that it is wise to select a video to a cluster seed if it contains a close enough time/GPS value, and two or three additional matching keywords. The better maximum performance gained in the tests gives a hint that lower weight values could be useful. This is because it allows more videos to be selected with only keyword information.

The sensitivity and specificity curves suggest that a threshold value in the region of 3 would yield optimal performance. The number of videos that enter the clusters should be at a reasonable level when the threshold is between 4 and 6.

Similar results can be seen in figures 16 and 17 for the extra data case. The difference is that in the curves, only small weights are used. This keeps the differences between the curves very small. The maximum performance gained is a little bit better in the extra data case than without extra data. Thus, extra data can help accuracy, but only a little at data set level.

The weighted majority voting yields similar results to the normal version of CA3. Thus, the use of a training data set for optimizing performance might not be needed.

6. Conclusions

The most important question is “will this kind of algorithm be useful in finding a set of videos from a certain event?” Based on the testing that we have done, we think the answer is “yes”.

There will always be a tradeoff between how many videos are returned as results, and their accuracy. The user can be assisted nicely by providing the probability scores.

The system could be improved by using other metadata that might be available: comments, web links, statistics etc.

The biggest challenges lie in the actual implementations of a video search system. The database has to be organized perfectly to make any reliable searches. This is because the amount of videos is so big that an exhaustive search is impossible.

7. References

1. Hila Becker, Mor Naaman, and Luis Gravano. Event Identification in Social Media. Twelfth International Workshop on the Web and Databases. 2009.
2. Calculate distance, bearing and more between Latitude/Longitude points. [Online] <http://www.movable-type.co.uk/scripts/latlong.html>.
3. Fast and effective text mining using linear-time document clustering. Larsen, B. and Aone, C. San Diego : Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, 1999.
4. Hila Becker, Mor Naaman, and Luis Gravano. Learning Similarity Metrics for Event Identification in Social Media. WSDM'10, February 4-6, 2010, New York City, New York, USA.
5. Shane Ahern, Mor Naaman, Rahul Nair, and Jeannie Yang. World Explorer: Visualizing Aggregate Data from Unstructured Text in Geo-Referenced Collections. JCDL'07, June 18-23, 2007, Vancouver, British Columbia, Canada.
6. Smitashree Choudhury, John G. Breslin, and Alexandre Passant. Enrichment and Ranking of the YouTube Tag Space and Integration with the Linked Data Cloud. The Semantic WebISWC 2009.
7. Junsong Yuan, Jiebo Luo, Henry Kautz, and Ying W. Mining GPS Traces and Visual Words for Event Classification. MIR'08, October 30-31, 2008, Vancouver, British Columbia, Canada.
8. Yi-Hsuan Yang, Po-Tun Wu, Ching-Wei Lee, Kuan-Hung Lin, Winston H. Hsu, Homer Chen. ContextSeer: Context Search and Recommendation at Query Time for Shared Consumer Photos.
9. Lyndon Kennedy and Mor Naaman. Generating Diverse and Representative Image Search Results for Landmarks. WWW 2008, April 21-25, 2008. Beijing, China.
10. Juan Ramos. Using TF-IDF to Determine Word Relevance in Document Queries. In Proceedings of the First First Instructional Conference on Machine Learning, 2003. Available: <http://www.cs.rutgers.edu/~mlittman/courses/ml03/iCML03/papers/ramos.pdf>
11. Wesam Ashour Barbakh, Ying Wu, and Colin Fyfe. Review of Clustering Algorithms. Non-Standard Parameter Adaptation for Exploratory Data Analysis. Volume 249/2009. Pages 7-28. ISBN978-3-642-04004-7. Springer Berlin / Heidelberg, 2009.
12. Teuvo Kohonen. Self-Organizing Maps. Springer, Berlin, Heidelberg. 1995.
13. Lawrence Hubert, Phipps Arabie: Comparing Partitions, Journal of classification 2:193-218 (1985), Springer-Verlag New York Inc.
14. Christopher Manning, Prabhakar Raghavan, and Hinrich Schutze. Introduction to Information Retrieval, Cambridge University Press. 2008. Available: <http://nlp.stanford.edu/IR-book/information-retrieval-book.html>

15. Phonetic Distance between Dutch Dialects. J. Nerbonne, W. Heeringa, E. Hout, P. Kooi, S. Otten, W.Vis. Antwerp : s.n., 1996. Proceedings of CLIN'95. ss. 185--202.
16. Wikipedia: Stop Words. http://en.wikipedia.org/wiki/Stop_words. 19.5.2010.
17. Richard O. Duda, Peter E. Hart and David G. Stork. Pattern Classification. 654 pages. John Wiley and Sons, 2000.

APPENDIX A: Missing Data Compensation Test Report

1 Introduction

This document is an appendix for the project report “Event Discovery by Clustering”. The purpose of the document is to give a short summary on how missing metadata fields “GPS” and “date” can be compensated by mining the text fields in the metadata.

In the GPS-case, free web services, Google Geocoding [A1] and Geonames [A2], were used for the task. Text from the metadata fields was fed to the services, and the given output was recorded to a database. This was problematic at times, because the web services had restrictions on how many requests could be sent in a day. Furthermore, there were network congestion problems that caused a small number of requests and GPS results to be lost.

In the date-case, we used a regular expression-based tool that we implemented.

The data set used for these tests was not the same that was used for testing the final versions of the clustering algorithms, but an older version with 1040 videos.

2 Text-to-GPS conversion

2.1 Dataset and algorithm

- 196 videos contained a text location
- 890 videos of the database were used for the statistics, all of which contained a ground truth for GPS location
- “Close enough” means here a distance of less than 20 km

Each text field is broken into a set of words. Each word is fed to a geocoding service, which gives GPS coordinates in return.

- No further preprocessing is applied at the moment

Figure 1 presents the principle of harvesting GPS coordinates.

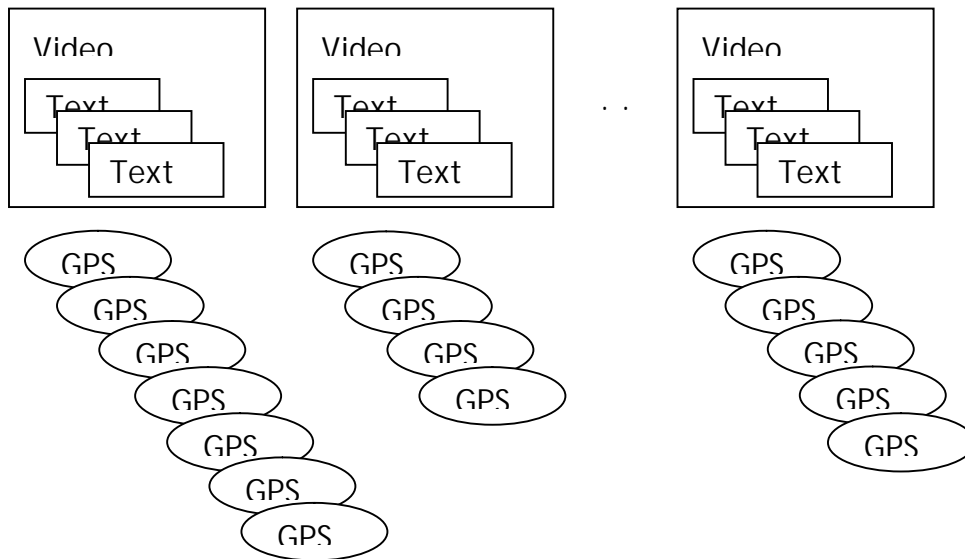


Figure 1 Principle for harvesting GPS coordinates

- A mean of 197 GPS locations were gathered for each video
- 1036 videos were given a GPS location ($887/890 = 99.7\%$)
- 660/890 videos did get a GPS location that was "close enough" e.g. points in the right place (<20km distance)
- 3.9% of all found GPS values were close enough
- The average percentage of close enough videos was 4.5% (for those videos that contained a close enough GPS coordinate)
- Mean of 10.3 close enough GPS coordinates were found if any
- Mean distance of found and ground truth GPS coordinates was 6252 km
- Median distance of found and ground truth GPS coordinates was 6195 km

2.2.2 LocationText

- A mean of 2.7 GPS locations were gathered for each video
- 171 videos were given a GPS location ($143/890 = 16.1\%$)
- 130/890 videos did get a GPS location that was "close enough" e.g. points in the right place (<20km distance)
- 46.7% of all found GPS values were close enough
- The average percentage of close enough videos was 49.0%
- Mean of 8.7 close enough GPS coordinates were found if any
- Mean distance of found and ground truth GPS coordinates was 321 km
- Median distance of found and ground truth GPS coordinates was 240 km

2.3 Google Geocoding ((normally 1 result per word))

2.3.1 All text fields (title, description, keywords, location)

- A mean of 13.2 GPS locations were gathered for each video
 - 879 videos were given a GPS location (879/890 = 98.7%)
 - 222/890 videos did get a GPS location that was "close enough" e.g. points in the right place (<20km distance)
 - 13.9% of all found GPS values were close enough
 - The average percentage of close enough videos was 22.4%
 - Mean of 2.3 close enough GPS coordinates were found if any
 - Mean distance of found and ground truth GPS coordinates was 5527 km
 - Median distance of found and ground truth GPS coordinates was 5711 km
-

2.3.2 LocationText

- A mean of 0.13 GPS locations were gathered for each video
 - videos were given a GPS location (58/890 = 6.5%)
 - 44/890 videos did get a GPS location that was "close enough" e.g. points in the right place (<20km distance) (MAXIMUM 31 videos because all 197 did not contain location text!)
 - 76% of all found GPS values were close enough
 - The average percentage of close enough videos was 88.5%
 - Mean of 1.9 close enough GPS coordinates were found if any
 - Mean distance of found and ground truth GPS coordinates was 15.7 km
 - Median distance of found and ground truth GPS coordinates was 14.9 km
-

2.4 Conclusions

- Geonames seems to find often coordinates that are close enough, whereas Google does not. This is because Google gives only one coordinate, while Geonames gives 10.
- LocationText gives much more accurate results than using all text fields
- A large amount of noise is present in all cases
 - Preprocessing should be used if all words are to be utilized

3 Text-to-DATE conversion

3.1 Description

- Objective
 - Find reliable time information from textual metadata
 - Data used
 - title, description, keywords
 - Algorithm
 - Search data for date formats using a template (regular expression)
 - Use heuristics to guess which date format is used
-

- Search data for other date expressions (month and year/day)
- Search date for year (numbers like 2005, 2009...)

3.2 Results

200 videos were used in the preliminary test. The algorithm did the following:

- DATE information
 - 47 videos were given a DATE (23.5% of the 200 videos)
 - 54 dates were given in total
 - 3 individual wrong DATEs were given
 - 1 double wrong DATE was given
 - 38, 4, 1 (single, double, triple) dates were correct
 - 12 available date information were not found (6% of the 200 videos)
- 43 correct, 4 wrong, 12 not found = 59 videos {73%, 7%, 20%}
- {91.5%, 8.5%} correct/wrong ratio of the found ones
- YEAR information
 - 94 videos were given a year (47% of the 200 videos)
 - 89 videos were given the correct year
 - 39,39,20 (single, double, triple)
 - 5 videos were given the wrong year
 - All year information was found
- {94.7%, 5.3%} correct/wrong ratio
- YEAR + month information
 - 4 year+month-pairs were found
 - All the found ones were correct
 - {100%}

3.3 Conclusions

- The text-to-date conversion works quite well, and the false positive ratio is acceptable.
- The conversion yields surprisingly many results
 - It is obvious that this kind of text crawling should be used for missing metadata search
- Better versions of the algorithm should be tested
 - Annotated bigger test set needed
- The algorithm cannot find well the month text + day/year pairs
 - Difficult due to language/representation differences
- Instead of using heuristics for guessing the date format used, all the different format options could be stored (like in the GPS case)

References

A1. Google Geocoding Service.

<http://code.google.com/apis/maps/documentation/services.html>

A2. GeoNames. www.geonames.org.