AUTOMATIC DETECTION OF SIGNIFICANT FEATURES
AND EVENT TIMELINE CONSTRUCTION
FROM TEMPORALLY TAGGED DATA

by

ABHIJIT ERANDE

B. E., University of Pune, 2005

A REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2009

Approved by:

Major Professor
William H. Hsu, Ph.D.

Abstract

The goal of my project is to summarize large volumes of data and help users to visualize how events have unfolded over time. I address the problem of extracting overview terms from a time-tagged corpus of data and discuss some previous work conducted in this area. I use a statistical approach to automatically extract key terms, form groupings of related terms, and display the resultant groups on a timeline. I use a static corpus composed of news stories, as opposed to an on-line setting where continual additions to the corpus are being made. Terms are extracted using a Named Entity Recognizer, and importance of a term is determined using the χ^2 measure. My approach does not address the problem of associating time and date stamps with data, and is restricted to corpora that been explicitly tagged. The quality of results obtained is gauged subjectively and objectively by measuring the degree to which events known to exist in the corpus were identified by the system.

Table of Contents

List of Figures

List of Tables

Acknowledgements

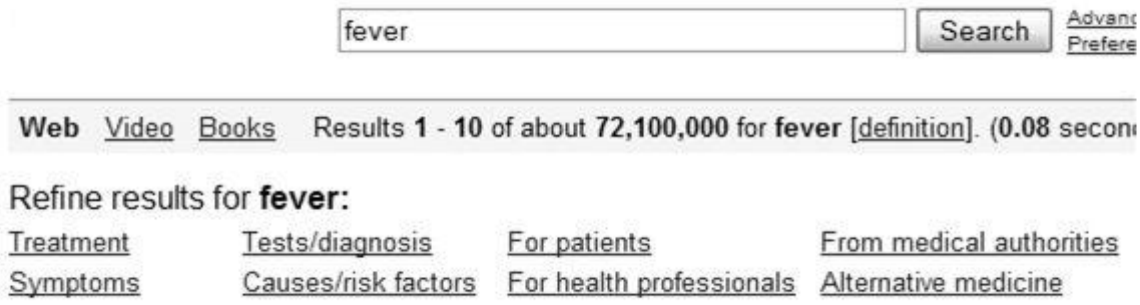
I would like to thank my advisor Dr. William Hsu, and the members of the Knowledge Discovery in Databases research group, without whose support this work would not have been possible.

CHAPTER 1 - **Introduction**

1.1 Background

This report addresses a text-based information extraction task known as event detection, the problem of identifying occurrences mentioned in text that are deemed significant or interesting according to some criterion. Event detection has applications to intelligent search, detection and tracking of trending topics from blogs and microblogs, and the application explored in this project: timeline construction from online news articles. Query-driven retrieval of information is useful if the topic on which further information is needed is clearly defined but cannot answer general queries like "What happened over the last month?".

The results returned by search engines are sorted using algorithms which prioritize results based on their popularity. A given search term may have different meanings in different contexts, and these alternate meanings may be overshadowed by results for more common usages of the search term. Sorting through this huge mass of data to identify the few hits of interest is a time consuming process and most people do not have the patience to scroll through page after page of results.



fever

Web Video Books Results **1** - **10** of about **72,100,000** for **fever** [definition]. (**0.08** secon

Refine results for **fever:**

Treatment Tests/diagnosis For patients From medical authorities
Symptoms Causes/risk factors For health professionals Alternative medicine

Figure 1.1: An ambiguous search term yields over 72 million hits, with interpretations ranging from a common aliment, to a music album by a popular artist

Results from some search terms, for example terms related to people, organizations, events and places , are strongly temporal in nature, and lend themselves very well to be viewed along a timeline. A timeline helps to visualize the order in which a search term has evolved over a period of time, and to get an idea of its significance at various points along the timeline. Furthermore, the timeline can also be annotated with additional relevant information, allowing for a surprisingly information-rich interface that at the same time is easy to comprehend.

1.2 Problem statement

The goal of this study is to make search results more accessible to users, by making the temporal aspect of the results more lucid. This is achieved by automatically extracting potentially important features from search results and clustering contextually related features together. The clusters of features so obtained are then displayed along a timeline in a way that makes the relative importance of features apparent.

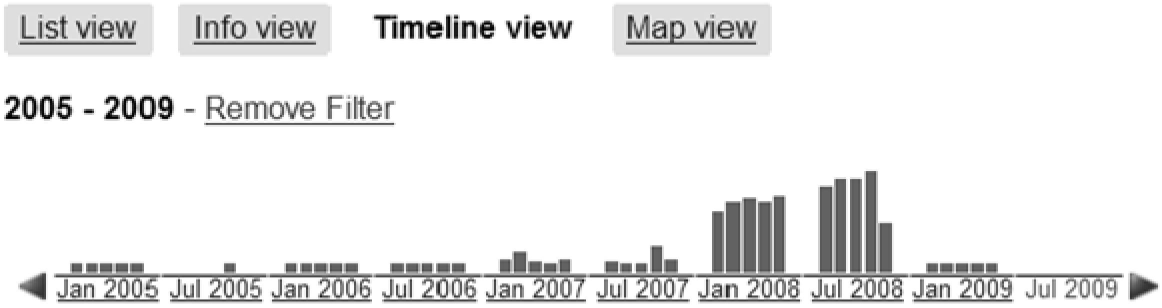


Figure 1.2***: A timeline view of search results for president Barack Obama. The vertical bars indicate the popularity of the search term for a specific time period***

We crawl the web for a particular search term to yield a corpus of pages for that particular term. We then extract features (named entities and noun phrases) from a corpus of documents resulting from the crawl results. Features are automatically extracted based on their perceived ‘importance’, a process described in Chapter 4. These extracted features correspond to significant events in the corpus, and are ranked based on their relative importance.

The process produces a ranked list of groups of features that correspond to significant events in the crawl results. Features determined are then grouped together if they are determined to be referring to the same event, and if they occur at roughly the same time. For each group we get a relative ranking of importance, a range of dates when it was important, and an indication of the amount of coverage in the corpus. This information is used to construct an overview timeline of the corpus, which displays related features as 'topics'.

1.3 Project objectives

I aim to develop a system that searches a corpus of date tagged news articles and automatically extracts features likely to be of relevance to users, where a feature is noun phrase or named entity. Relevance judgments are made by statistically determining if the appearance of a feature is random or not.

A list of relevant features so obtained is likely to have multiple features that refer to the same event. I then look for co-occurring features (features with a high degree of overlap in their date ranges) We make the initial assumption that two co-occurring features are not related, and use a χ^2 test to distinguish random association from true association. Once features that are related have been found, they are grouped together into what I call 'topics', and a date range or ranges for each topic is determined.

Finally, I aim to construct a timeline using the SIMILE API and display the topics found using the method described above.

1.4 Project methodology

First, I present a literature review of previous and current attempts at event detection and clustering, where I aim to identify relevant research. Once relevant work has been found, I aim to determine the pros and cons of each individual approach and determine if existing work can be adapted or expanded upon.

Next, I describe how I selected a group of date-tagged news articles pertaining to a limited set of events that have received significant coverage in the news. These articles will be then be run though the Stanford Named Entity Recognizer in order to determine list of Named Entities and Noun Phrases contained in each individual article.

Once lists of features have been obtained, relevance judgments will be made for each feature, in each article. Co-occurring Features will be then tested for association and features determined to be referring to the same event will be grouped into topics. Each topic will have a date range, or multiple date ranges for which it was important calculated. This information is used to create a timeline using SIMILE.

In order to determine the efficacy of the system, the topics identified by the system are manually compared to a list of important topics known to exist in the data set. Also, subjective judgments about the quality of the search results will be made by end-users of the system.

CHAPTER 2 - **Background**

2.1 The need for a timeline interface

Today, users of search engines are presented with results that may run into millions of pages. It is extremely difficult for users to sort through a huge mass of results and find hits corresponding to potential topics of interest. For example, consider someone who has returned from a couple of weeks of vacation without access to a news source and who now wants to know what has transpired during his absence. He would have to go through two weeks' worth of newspapers, but this is a potentially time consuming proposition. An automated information-extraction system could assist him by automatically picking out topics that have lately received significant coverage in the news, and bringing them to his attention. Similarly, consider an analyst whose task is to monitor the web for news of disease outbreaks from all over the world, and to determine if a disease outbreak in some part of the world has the potential to become more widespread. In such a usage scenario, it would be helpful if a system was available that would search for news reports that contain references to a pre-programmed list of diseases of interest. If the number of reports from any area that mention a disease cross a threshold, the system would automatically bring the situation to the attention of the analyst.

2.2 Result visualization

Several attempts have been made to improve the presentation of search results to users, usually by attempting to rank search results by importance. The most famous of these methods is the PageRank algorithm (Page, Brin, & Motwani, 1999), which attempts to measure the importance of each page in a set of linked documents. PageRank is a link analysis algorithm used by the Google Internet search engine that assigns a numerical weighting to each element of a hyperlinked set of documents, with the purpose of "measuring" its relative importance within the set. The algorithm may be applied to any collection of entities with reciprocal quotations and references. PageRank uses the link structure of the internet as an indicator of an individual page's value. Essentially a link from page A to page B is interpreted as a vote by page A, for page B. The algorithm also considers factors other than the volume of votes, or links a page receives; it also analyzes the page that casts the vote. Votes cast by pages that are themselves "important" weigh more heavily and help to make other pages "important". However, with respect to the scenarios outlined above, this approach is not conducive to identify overall trends in the data. This is because the algorithm simply returns a ranked list of results, and does not take into account the temporal aspects of the results or the relationships between entities present in the search results.

On the other hand, graphical user interfaces such as timelines are simple and intuitive and increase the accessibility of information to a wider audience. A timeline exploits spatial and visual clues to provide a graphical representation that is more natural and closer to innate human capabilities. Spatial relationships are understood more quickly than verbal representations, and visual thinking is believed to be quicker than logical thinking (Galitz)

There have been a number of systems built for the purpose of browsing the information within large collections of data. These systems select significant words and phrases, and display them in a manner that allows the user to graphically gist the significant topics in the collection. Examples include I^3R (Croft & Thompson), where the knowledge base is displayed graphically as a network of nodes and links. Nodes represent entities such as documents, and links represent relationships between the entities. Wise, *et al.* (1995) discuss representations called ThemeScapes and Galaxies, where ThemeScapes are abstract, three-dimensional landscapes of information that are constructed from document corpora, and the Galaxies visualization which displays cluster and document interrelatedness by reducing a high dimensional representation of documents and clusters to a 2D scatterplot of ‘docupoints’ that appear as do stars in the night sky. Kohonen (2001) developed the Self Organizing Map, a type of artificial neural network-based unsupervised learning model that can be applied to vector representations of text documents to produce a similarity graph of input data. These systems are all term, rather than document centered, and none of them makes explicit use of time.

2.3 Previous work on event extraction

(Swan & Allan, 2000) and [Yang, Pierce, Carbonell] are primarily concerned with the methodology of automatically selecting features from a corpus for display. (Swan & Allan, 2000) also discusses the use of timelines as a browsing interface to a large collection of documents, and largely build upon the work described in (Swan & Allan, 1999)

The authors make use of the TDT-2 dataset and require a corpus that has already been time-tagged. Features are identified and statistically significant features are extracted (These correspond to objects or events that could be of potential interest to end users of the system). To obtain the list of features, a shallow parser was used to obtain noun-phrases, and a named entity extractor was used to find locations, organizations, and names of people. The named entities and noun-phrases so obtained are what the authors use as 'features'. Once features have been extracted, they are ranked based on how likely they are to have a high content bearing. Once features have been extracted, they are grouped onto clusters. Clustering is performed on the notion of 'topic', as defined by the TDT studies. The groups of features so obtained are used to automatically create an interactive timeline view that displays the major events and topics contained in the corpus of data.

The TDT-2 dataset is a collection of 21,255 documents containing 192 topics with known relevance judgments. In order to produce groups of related features, the system begins by generating a list of named entities and noun phrases contained in the dataset. It then divides the corpus into days, and calculates the number of documents containing a feature on any given day. Statistical significance of features is calculated using the χ^2 metric. It is assumed by default that there is no correlation between features, unless co-occurrence is shown to be shown to be above a level of significance. Using the number of documents for any given day, the number of documents for that day containing a feature in question, the total number of documents in the corpus, and the degree of freedom for that feature, the χ^2 value can be calculated. The χ^2 value is only calculated if the occurrence on that day is more than what would be predicted by chance.

The χ^2 value is compared to a predetermined threshold, and runs for consecutive days over the threshold are combined into a single range. Next, a measure of how distinctive the feature was at its peak value is calculated. This is determined by calculating the χ^2 value for every sub range of the range under consideration, and choosing the highest value thus obtained. Terms with significant appearances in the corpus and their associated ranges are then selected and sorted on the maximum χ^2 value. This yields a sorted list of the most significant features in the corpus and their dates.

These groups of features are then clustered into topics by selecting the highest-ranked unclustered feature and comparing the time ranges with all lower ranked features. If the dates overlap, a χ^2 calculation is performed, and if it is over a predetermined threshold, the feature is marked as a potential member of the cluster. Once the list of features has been processed entirely, a standard hierarchical agglomerative clustering on the marked features is performed. The dendrogram is then cut at a predetermined threshold, and the cluster containing the original central element is taken as the valid cluster. Average link clustering was used, as it tended to produce uniformly good results while being tolerant of minor weighting errors.

Each cluster obtained was assigned a cluster name consisting of the highest ranked named entity followed by the highest ranked noun phrase. Additionally, the following attributes were associated with each cluster:

1. **Importance:** A relative ranking of importance of the cluster
2. **Range:** The range of dates for which the cluster was important
3. **Coverage:** An indication of the amount of coverage received by the cluster in the corpus
4. **Interestingness:** A measure of how distinctive or surprising the cluster is
5. **Term count:** The number of distinctive search terms that are associated with that topic

Finally, the timeline was then constructed using the cluster information determined above. In my paper, I only make use of the Importance and Range attributes.

[Allan, Papka, Lavenko] discusses the problems of detecting new events, and tracking existing events in a stream of news stories. New event detection entails identifying new stories that discuss an event that has not been reported in previous stories, while event tracking refers to finding all subsequent stories that are related to a few seed stories.

An event is defined as something that happens at a particular time and place. In order to determine if two events are the same, the authors introduce the concept of event identity, which is the set of properties that makes two events the same.

The data set used in this study was the TDT corpus, which contains 15,863 news stories and 25 events. To establish a benchmark for the purpose of evaluating the effectiveness of the system, every story was judged with respect to every event. Effectiveness of the system was measured by the miss (false negative) and the false alarm (false positive or fallout) rates. A miss occurs when the system fails to detect a new event, and a false alarm occurs when the system indicates that a story contains a new feature, when it does not. A Detection Error Tradeoff curve (Martin) is used to show how false alarm and miss rates vary with respect to each other at various threshold values.

The new event detection algorithm is a modification of the single pass clustering algorithm described in (Rijsbergen). It processes new stories on-line (as they arrive) as follows:

- Use feature extraction and selection to build a query representation of the story's content.
- Determine the query's initial threshold is by evaluating the new story with the query.
- Next, compare the new story with earlier stored queries

- If the story triggers no previous query by exceeding its threshold, flag the story as containing a new event, otherwise, if an existing query is triggered, flag the story as not containing a new event.
- Add the story to the agglomeration list of queries that it triggered
- If needed, rebuild existing queries using the story
- Add the new query to memory.

In order to evaluate the system, the authors carried out a subjective evaluation and an objective evaluation. A subjective evaluation carried out by persons other than the authors deemed the topics formed by the system to be reasonable.

In order to carry out an objective evaluation, a text narrative of the major news stories of the year called 'Facts on File' was used. The list of stories from Facts on File was taken and reduced to a machine readable form, where a date range for each story was given and a list of noun-phrases and significant names that might be found were listed. Stories from Facts on File were considered as relevant, and stories not listed were considered irrelevant. The output of the system was compared with the list of stories from facts on file, and clusters identified by the system were deemed relevant if the date range of the cluster corresponded to the date of the story, and if there was at least one feature in common with the derived story and the judged story.

2.4 The SIMILE timeline project

SIMILE timeline is a DHTML based AJAX widget. SIMILE allows users to easily create graphical representations of a chronological sequence of events. The purpose of the project was originally to create a tool for visualizing a schedule of activities, but over time has evolved to become much broader in scope. The method described in this paper makes extensive use of SIMILE's ability to represent events that take place over a period of time, as opposed to discrete events.

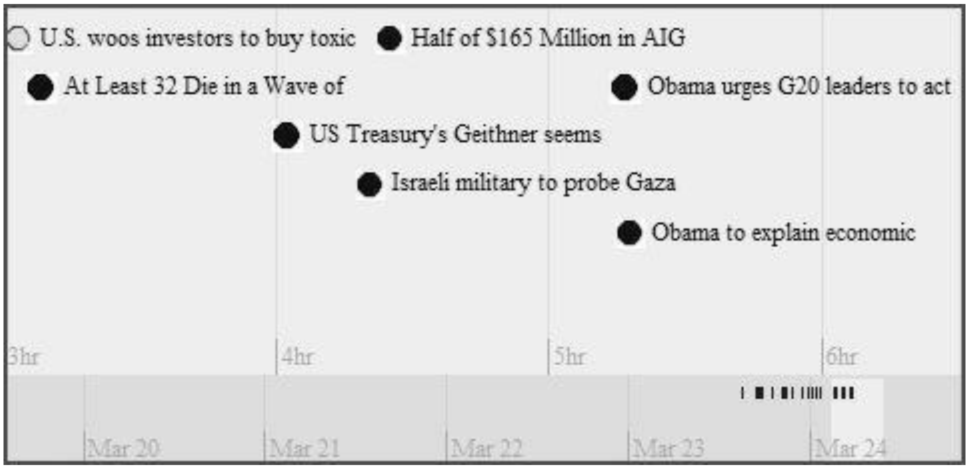

U.S. woos investors to buy toxic
Half of $165 Million in AIG
At Least 32 Die in a Wave of
Obama urges G20 leaders to act
US Treasury's Geithner seems
Israeli military to probe Gaza
Obama to explain economic
3hr
4hr
5hr
6hr
Mar 20
Mar 21
Mar 22
Mar 23
Mar 24

Figure 2.1: ***A portion of a SIMILE timeline showing news events over a three hour time period. The upper band displays hourly news, while the lower band displays events taking place over a period of several days.***

A timeline contains one or more bands, which can be panned infinitely by dragging with the mouse pointer. A band can be configured to synchronize with another band such that panning one band also scrolls the other. Bands show the same events at different resolutions, for example, the bottom band can show events over a period of several years, while the upper band provides an expanded view of a small section of the lower band.

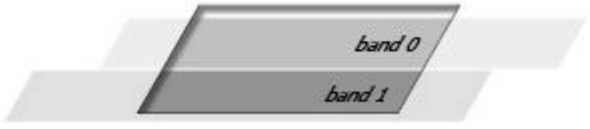


Figure 2.2*: Bands*

A band is responsible for supporting panning as well as coordinating its various sub-components:

- An ether, which maps between pixel coordinates and dates/times. It specifies how many pixels are taken up by a time span.
- An ether painter, which paints date/time labels (or other markings) and the background of the band as well as the highlight (the lighter part of the lower band in the first timeline above)
- Zero or more decorators, which further decorate the background of the band.
- An event painter, which paints the events.

The band also takes an event source which provides events to be displayed in that band. Different bands can have different event sources. This flexibility allows for timeline mashups. Various sub-components that do painting take a theme, which stores default visual and behavioral settings.

A timeline is implemented as a div element that contains inner div elements as its bands. The band divs are cropped and positioned relative to the timeline div. A band div itself contains several inner elements that implement various parts of the band. The bands also have different background colors, and the weekly band of the second timeline has weekend markings. All of these visual elements are "painted" by adding HTML elements to the band divs at the appropriate positions.

As a band is panned, its div is shifted horizontally or vertically, carrying all of its visual elements along. When either end of the band div approaches the visible (non-cropped) area, the band div is re-centered, its coordinate origin is changed, and then its various visual elements are re-"painted" relative to the new coordinate origin. All of this "paging" is done as seamlessly as possible so that the user experiences smooth, infinite panning.

CHAPTER 3 - **Methodology**

3.1 Overview

The system finds named entities and noun phrases (features) that are likely to have a high content bearing, as determined by conducting a χ^2 test. Features are extracted from time-tagged news articles, and are marked with the date of4 the news story that they were extracted from. Features that stay important for a number of consecutive days are consolidated into a date ranges for that feature. Date ranges for multiple features are compared to determine if any overlap exists. If features with overlapping ranges are found, it is likely that they both refer to the same event in the news, and the features are consolidated into a group.

For years, a Congressional hearing with Alan Greenspan was a marquee event. Lawmakers doted on him as an economic sage. Markets jumped up or down depending on what he said.

From this example, the following features will be extracted; and ranked based on their likely importance in the context as follows:

1. Congressional hearing
2. Alan Greenspan
3. Markets
4. Lawmakers
5. Sage

Next, features are clustered on the notion of topic. The result of this will be:

- *Alan Greenspan = {Congressional hearing, Alan Greenspan}, importance = 1*

- Markets = {markets} *, importance = 2*
- Lawmakers = {lawmakers}, *, importance = 4*
- Sage = {sage} *, importance = 4*

The group information extracted in this way is then projected onto a timeline, an example of which can be seen in the figure below:

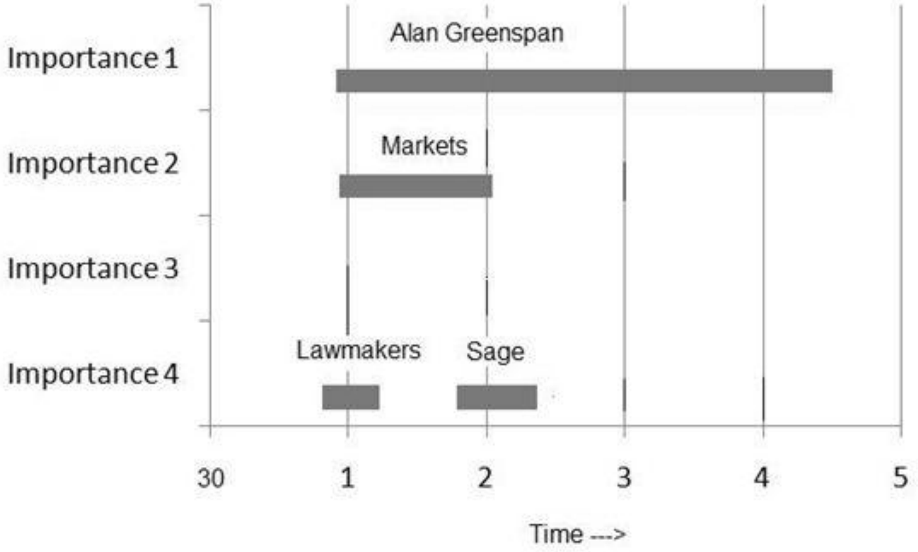


Figure 3.1***: Timeline view of identified topics***

3.2 System overview

The system can be broadly divided into several distinct components. A broad overview of the components involved is as shown in Figure 3.2 below:

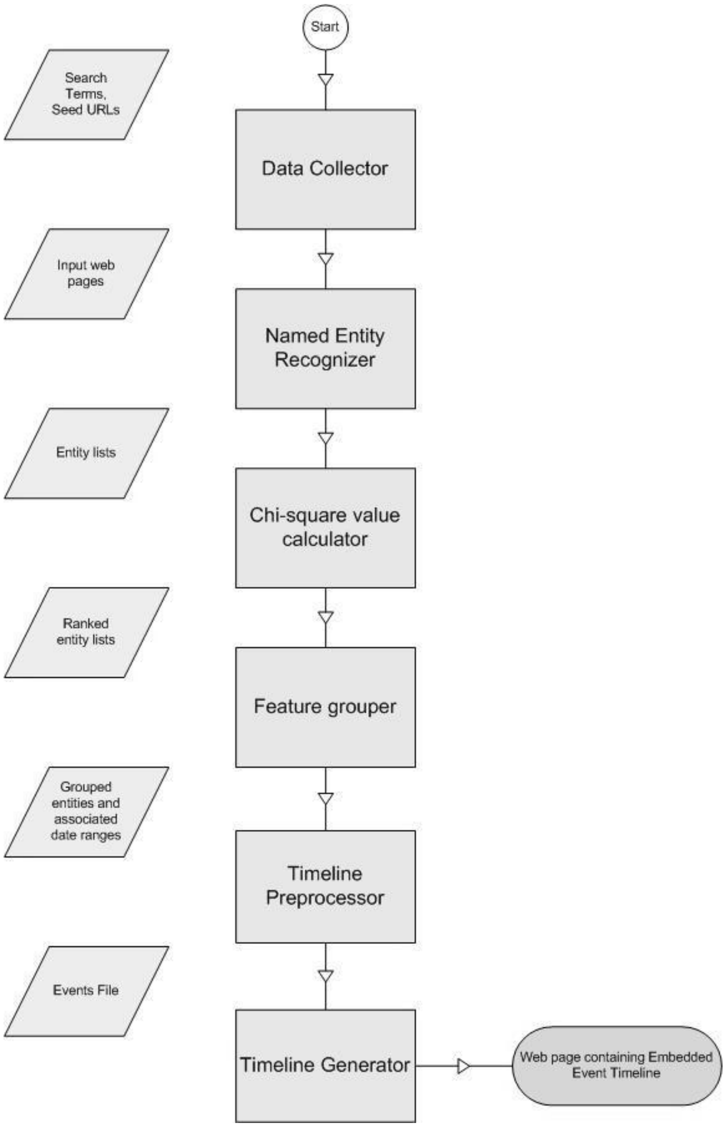


Figure 3.2***: System overview***

Data Collector: The data collector uses the Heritrix web crawler and a list of pre-specified seed terms and search URLs to obtain the input data set. As the approach described in this report relies on the presence of explicit date stamps for individual articles, the web crawler was run on the web site of the Reuter's news reporting agency, which tags each story with a date and location, amongst other attributes.

Named Entity Recognizer: The named entity recognizer is run on the results of the web crawl, obtained from the data collector. This report makes use of the Stanford NER. This is a four class NER tagger, and divides the extracted entities into the following classes:

PER: Names of persons

LOC: Geographical locations

ORG: Entities such as government organizations, institutions

OTHER: Any entity that cannot be classified as any of the preceding three

Each entity extracted is tagged with the date of the article in which it was found.

Chi-Square Calculator: The Chi-Squared value of a feature provides a measure of how distinctive the feature under consideration is. This part of the system takes as input the list of entities found by the named entity recognizer, and calculates for each feature a Chi-Square value. This value is calculated for every day in the corpus and for every term found on any given day. Features below a pre determined significance level are discarded.

Feature Grouper: Due to the nature of news stories, many of the features detected by the system co-occur with each other, and refer to the same event. In order to avoid displaying multiple timelines for a single news event, features that belong to one event are identified and grouped together to avoid visual clutter on the timeline. Additionally, events that span multiple days are identified, and date ranges are calculated for these events.

Timeline Preprocessor: In order to create a timeline, a SIMILIE recognizable events file needs to be created. This file controls formatting instructions, centers and adjusts the scale of the timeline and controls the display of events. This part of the system takes in the list of groups of events and individual features with their corresponding date ranges, and generates an XML file that can be recognized by SIMILIE.

3.3 Identifying significant features

We begin by generating a list of all the named entities and noun phrases (locations, organizations, names of people) present in the corpus. It is assumed that features are produced as a result of a random process with an unknown binomial distribution. Further, we assume initially that there is no association between features, i.e. the co-occurrence of two features is devoid of meaning until it is shown to be statistically unlikely. We are interested in determining if the appearance of a feature is random or not. Features shown to be not random can be considered as 'interesting' and processed further.

The χ^2 statistic is used for measuring the strength of association, as it is an excellent statistic for distinguishing random association from true association. We begin by dividing the corpus into individual days. Next, for every day in the corpus, we generate a list of features occurring on that day. We discard all features that have four or fewer occurrences in the corpus.

In order to perform the χ^2 test, we need to define what we are taking as samples, and what we are taking as occurrences. Here, we take samples as documents, and define an occurrence as any document that contains one or more instances of a feature under consideration. This statistic is referred to as *df* (document frequency).

a. The number of documents containing the feature for the current day
b. The number of documents not containing the feature for the current day
c. The number of documents containing the feature over the entire corpus
d. The number of documents not containing the feature over the entire corpus

	f_j	$\bar{f_j}$
$t = t_0$	a	b
$t \neq t_0$	c	d

Table 1*: Contingency table for calculating the χ^2 value of individual features*

Knowing the number of documents from a given day, the number of documents on that day containing the feature (f_i), the total number of documents in the corpus (N), and the number of degrees of freedom (df) for the feature, we can form a 2 x 2 contingency table. This is modeled by a χ^2 distribution with one degree of freedom. Using Table 3.1, we can obtain χ^2 from the following equation:

$$\chi^2 = \frac{(ad-bc)^2}{(a+b)(a+c)(b+c)(b+d)}$$

The χ^2value is calculated for every feature in every article from the corpus. The χ^2 threshold is set at 7.878, which corresponds to a probability of 0.005 that a feature from a stationary process would identified as being random. In other words, this would yield 5 false hits a day in a 1000 event corpus: a sufficiently low value. Additionally, for events occurring over consecutive days, this probability is even lower. Runs of consecutive days above this threshold are combined into a single range. If the χ^2 value is above 7.879, the feature is considered to be significant and is tracked further.

We may obtain multiple disjoint date ranges for some features. In this case, runs separated not by more than one day are combined into a single range. To calculate a measure of how distinctive a feature is at its peak value, the χ^2 value is calculated for every subrange of the range under question, and the highest value is chosen.

3.4 Grouping similar features

The features and their associated ranges that we have identified are produced by news stories and events. For example consider 3.1, which contains Pirates, Somalia and Maersk Alabama, all of which are terms from the same story (note the overlap in dates). For a given event there are usually multiple terms that are associated with it. Grouping these terms together reduces the total number of events that must be comprehended, and makes these evens easier to identify.

After selecting terms with significant appearances in the news, and associated ranges, we sort on their significance (their χ^2 value). This yields a list of the most significant events in the corpus and their dates. These features are then grouped into topics by taking the highest ranked ungrouped feature, and comparing the time ranges with all lower ranked ungrouped features. If the date ranges for two features overlap, we test the default assumption that these features are independent over the time span in question. If the test shows that independence is statistically unlikely, we mark these terms as related. This is done by performing a χ^2 calculation and if the value is above a threshold, this feature is marked as a potential member of the cluster.

Event	**Date Range**
Pirates	April 07 - 13
Chrysler Corporation	April 20 - 22
Somalia	April 09 - 13
Maersk Alabama	April 07 - 13
Swine Flu	April 24 - 25

Table 2*: Date ranges for top ranked features by χ^2values*

As an example, consider the terms in Table 3.1. We do not consider overlapping Maersk Alabama with Swine Flu as the date ranges do not overlap. However, Pirates does overlap, so we consider the chi-square value for the pair of terms for that date range. That value is 541.2, which is well over our threshold, so they are merged. The next term that overlaps is Somalia, with a score of 149.7, so we merge that as well. This process continues until no more terms can be merged.

In order to group related features together, we sort the features by their χ^2 values. For each feature, we compare its date range with that of lower ranked ungrouped features. If there is an overlap in the date range, we test the hypothesis that these features are independent by invoking a second association. The assumption that two features f_i and f_k have independent distributions implies that $P(f_i)= P(f_j|f_k)$. This is tested for the time spans where features f_j and f_k are significant. The resulting counts form a 2 x 2 contingency table where:

a. The number of documents in a given time span where f_k and f_j co-occur
b. The number of documents where f_j occurs without f_k
c. The number of documents where f_k occurs without f_i
d. The number of documents containing neither feature

	f_j	$\bar{f}_j$
f_k	a	b
$\bar{f}_k$	c	d

Table 3*: Contingency table for calculating the χ^2 value in order to determine co-occurrence*

If the χ^2 value calculated in this way lies above our threshold of 7.879, we conclude that the features are related and add them to the group.

CHAPTER 4 - **Experiments**

4.1 Evaluation criteria

Information Retrieval Systems are usually judged by determining if a system's results (on a fixed set of queries on a fixed corpus) are relevant or not. These relevant judgments are made by human assessors. A subjective assessment showed that the system-generated topics created by grouping features tend to be of high quality for an automatic system, i.e., most of the retrieved features are reasonable and .

For objectively determining overall usability of the system, including that of the timeline GUI interface, it is proposed to obtain reviews of the system from persons other than the author.

4.2 Corpus retrieval

The dataset used for my experiments consisted of a set of about 120 news articles collected over a period of several contiguous days from the news reporting agency Reuters. The dataset was obtained using a modified version of the Heritrix webcrawler. The crawl was seeded with the following groups of terms, each of which had received substantial news coverage at the time of running the crawl.

1. Pirates, Somalia, NATO, Navy
2. Swine Flu, Outbreak, Mexico, WTO
3. General Motors, Chrysler, Bankruptcy, ATF

The advantage of using news articles that they are already tagged with the date on which the event occurred. Automatically extracting dates from web pages is possible, but in practice a page may contain references to other days that we are not interested in. For example, consider this snippet from a news article which illustrates the difficulty in identifying the correct date associated with an event:

An Italian cruise ship used guns and a fire hose to beat off a pirate assault. A South Korean tug boat with 16 crew onboard, is still being held in northern Somalia after it was seized on April 11.

This article is about the event involving the Italian ship, and is dated 29th April, but contains reference to an incident involving a Korean ship that took several days previously. In this case, we need to make a relevance judgment and select the correct date. Using a pre tagged corpus sidesteps this problem.

4.3 Entity recognition

The corpus so obtained using the crawler was partitioned into individual days. The Stanford Named Entity Recognizer was then run on the articles from each day in the corpus, and used to generate a list of Named Entities (People, Organizations, Locations) for that day. Each named extracted is considered to be a feature of the news article.

The dataset was broken up into days because as described in the previous section, I make use of the *df* (document frequency) statistic.

4.4 Chi-square value calculation

After generating lists of features for each day, we calculate the number of documents containing a particular feature on each day of the corpus and the total number of documents containing the feature in the corpus. With these numbers, and knowing the total number of documents in the corpus, the χ^2 value of each feature for each day is calculated. The χ^2 value for a feature on a particular day in the corpus is only calculated if it is contained in three or more documents from that day.

The conventionally accepted significance level is 0.05 or 5%. This corresponds to a χ^2 value of 3.841, for a distribution with one degree of freedom. The next step is hence to discard all features that have a χ^2 value that is below the threshold.

4.5 Experimental results: A case study

In order to determine if the appearance of a feature, a χ^2 test is performed for every feature on every set of documents comprising a day. For the χ^2 test, we need to define what are taken as samples, and what are taken as occurrences. I use a statistic known as df (document frequency), where samples are documents, and an occurrence is any document that contains one or more occurrences of a feature under consideration.

Statistics are only calculated for features with df > 2. For each feature, and for each date, the χ^2 value is calculated. If it is above 3.841, which corresponds to a probability of 0.05, we begin tracking the feature. The largest contiguous block of days where for each day the feature was significant is assembled. For example, the χ^2 values for the feature *U.S. Centers for Disease Control and Prevention,* starting with April 23 are as shown in Table 4.1. From April 23th to April 26th, the feature has a χ^2value of > 3.841, hence the entire date range is associated with this feature.

April 23	April 24	April 25	April 26
4.78	**24.01**	**4.78**	**4.83**

Table 4*:* ***χ^2values for the feature "U.S. Centers for Disease Control and Prevention"***

The features and their associated date ranges are produced by news stories and events. Every news story tends to have a number of features associated with it, hence in order to simply matters for the end users, features associated with the same news story are grouped together. Grouping features reduces the number of objects displayed on the timeline, and makes news stories easier to identify. Table 4.2 shows some of the χ^2 values found by the system, and their associated date ranges.

Feature	Chi-Square value	Dates
Chapter 11	45.51724138	23, 24, 25
Pakistan	35.0798419	23
Swat Valley	35.0798419	23
General Motors	30.31468531	23, 25
Chrysler	30.31468531	23, 24, 25
Swine Flu	30.31468531	23, 24, 25, 26
Mexico	30.31468531	23, 24, 25, 26
California	10.90909091	23, 24, 25

Table 5*: Features ranked by* χ^2 *values*

In order to group stories, the feature list is first sorted on theχ^2values. For each feature not part of a group, the date range for the feature is compared with that of lower ranked features. If there is an overlap in date ranges, the default assumption that features are independent is tested by carrying out a χ^2test. If the test shows that independence is statistically unlikely, the features are marked as related. For example, consider the entries in Table 4.2. It can be seen that *Chapter 11* overlaps with *Chrysler*, and when the χ^2values for the two features are calculated, they are found to be above the threshold. Hence the two features are merged. The next feature that has a date overlap is *General Motors*, which is also above the threshold and it too is merged. After merging terms, we obtain the news stories shown in Figure 4.3.

News Story	**Date Range**
General Motors, Chrysler, Chapter 11	April 23 - 26
Swine Flu, Mexico, California	April 23 - 26
Pakistan, Swat Valley	April 23

Table 6*: Extracted news stories*

The news stories and their date ranges so obtained are then written to an events file that is in a format that can be recognized by the SIMILIE timeline generator, along with instructions that specify the date range and resolution of the timeline. SIMILIE automatically takes care of how events are laid out and produces a timeline.

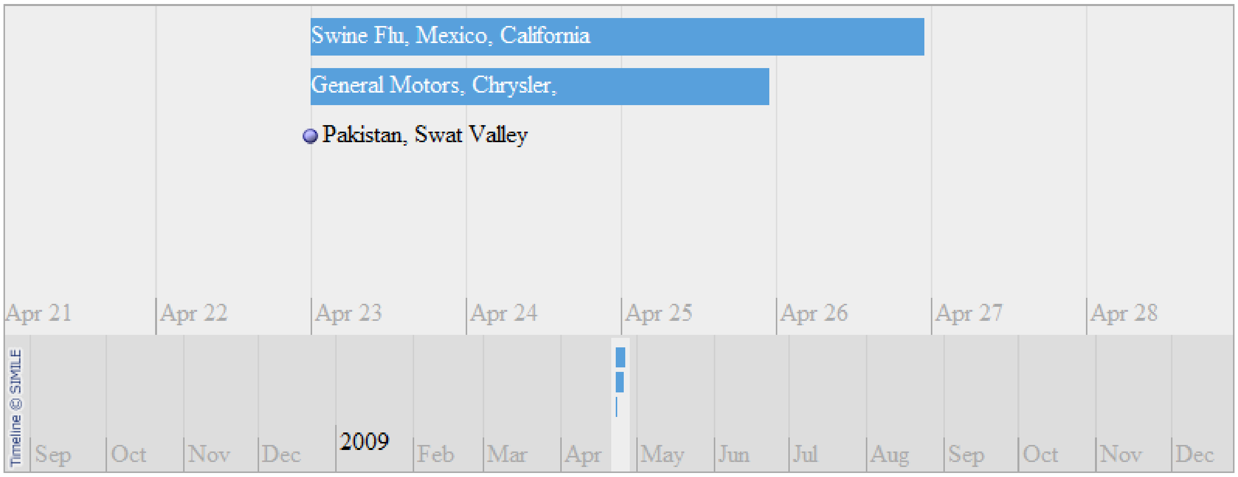


Figure 4.1*: Constructed SIMILIE timeline*

CHAPTER 5 - **Conclusion**

This report presents a technique for generating clusters of named entities and noun phrases that capture the information corresponding to major news topics covered in the corpus. The resultant clusters were evaluated with the help of human assessors, who felt that the resultant groupings of features were very indicative of important topics within the dataset

Ultimately, I would like to implement a system that automatically tags events with dates, rather than relying on an explicitly time-tagged corpus. In the future, the semantic web will allow for tagging of a large amount of information, rather than just the date to be associated with web pages. This metadata could conceivably be used to garnish the timeline with other relevant information. It is not hard to imagine a use case scenario where clicking on an event on the timeline allows for viewing additional information about that event, all of it gleaned automatically from the source document.

Future improvements to the system could focus on improving the accuracy of feature extraction. At the moment, about 1 in every 200 features extracted for single-day events is spurious although events spanning multiple days are far less susceptible to this kind of error.

The techniques presented in this report can make a significant contribution to the accessibility of information, as it allows the creation of an overview timeline that provides a high-level overview of the content of a large amount of data. As the amount of metadata available online increases, systems similar to the one described here can be expected to become more common, thus simplifying user interaction.

References

- Croft, W. B., & Thompson, R. H. (2007). I3R: A new approach to the design of document retrieval systems. *Journal of the American Society for Information Science* , 389 - 404.
- Galitz, W. O. (1996). *The Essential Guide to User Interface Design.* Hoboken, NJ: John Wiley & Sons.
- Kohonen, T. (2001). *Self-organizing maps.* New York, NY: Springer.
- Page, L., Brin, S., & Motwani, R. a. (1999). *The PageRank citation ranking: Bringing order to the Web.* Palo Alto, CA : Stanford University Press.
- Swan, R., & Allan, J. (2000). Automatic Generation of Overview Timelines. *Conference on Research and Development in Information Retrieval.* Athens, OH, USA: ACM Press.
- Swan, R., & Allan, J. (1999). Extracting Significant Time Varying Features From Text. *Conference on Information and Knowledge Management.* Kansas City: ACM Press.
- Wise, J., Thomas, J., Pennock, K., Lantrip, D., Pottier, M., Schur, A., et al. (1955). Visualizing the Non-Visual: Spatial analysis and interaction with information from text documents. *IEEE Symposium on Information Visualization* , 55.

Appendix A: $\chi 2$ tests

A $\chi 2$ test (chi-square test) is any statistical hypothesis test in which the test statistic has a chi-square distribution when the null hypothesis is true, or any in which the probability distribution of the test statistic (assuming the null hypothesis is true) can be made to approximate a chi-square distribution as closely as desired by making the sample size large enough.

An example of where the distribution of the test statistic is an exact chi-square distribution is the test that the variance of a normally-distributed population has a given value based on a sample variance. Such a test is uncommon in practice because values of variances to test against are seldom known exactly.

Some examples of chi-squared tests where the chi-square distribution is only approximately valid are:

- Pearson's chi-square test
- Yates' chi-square test
- Mantel-Haenszel chi-square test.
- Linear-by-linear association chi-square test.
- The portmanteau test in time-series analysis, which tests for the presence of autocorrelation
- Likelihood-ratio tests in general statistical modeling, for testing whether there is evidence of the need to move from a simple model to a more complicated one (where the simple model is nested within the complicated one).

Appendix B: The χ2 distribution

The χ2 distribution (or chi-square distribution) is one of the most widely used theoretical probability distributions in probability theory and inferential statistics. It is useful because, under reasonable assumptions, easily calculated quantities can be proven to have distributions that approximate to the chi-square distribution if the null hypothesis is true. In this paper, it is used to test if the co-occurrence of two events is statistically significant.

The best-known situations in which the chi-square distribution is used are the common chi-square tests for goodness of fit of an observed distribution to a theoretical one, and of the independence of two criteria of classification of qualitative data. Many other statistical tests also lead to a use of this distribution, like Friedman's analysis of variance by ranks.
Definition:

If X_i are k independent, normally distributed random variables with mean 0 and variance 1, then the random variable:

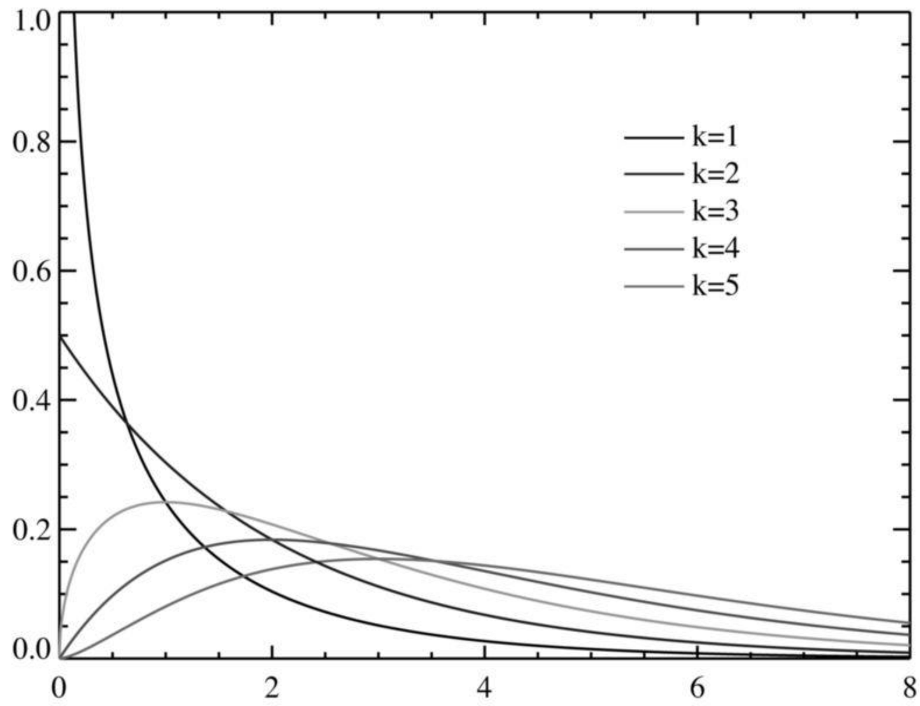
$$Q = \sum_{i=1}^{k} X_i^2$$

is distributed according to the chi-square distribution with k degrees of freedom. This is usually written as:

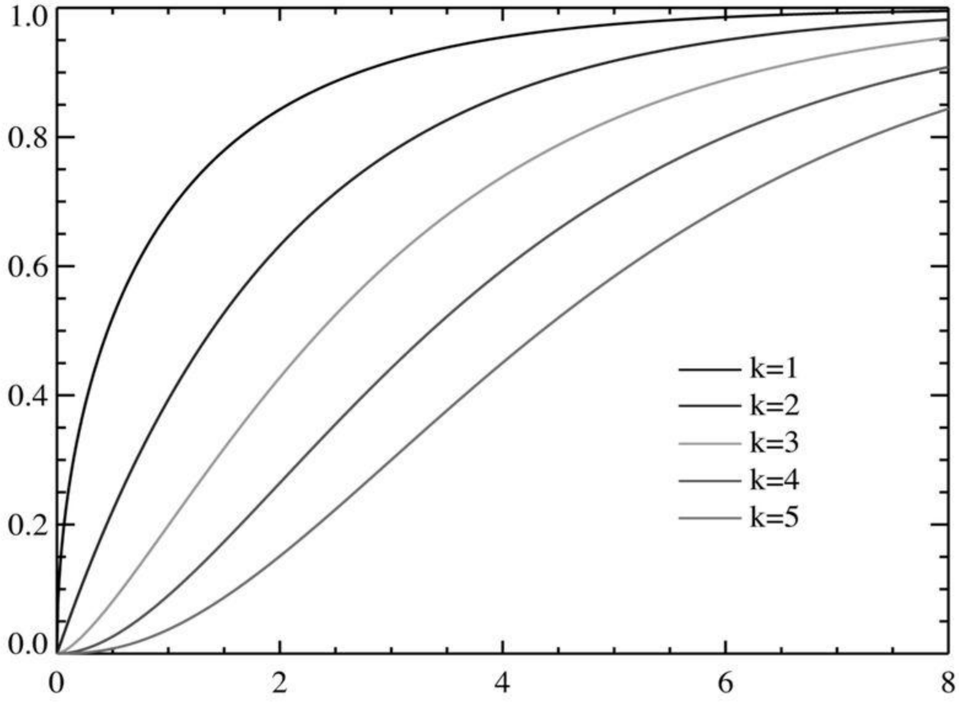
$$Q \sim \chi_k^2.$$

The chi-square distribution has one parameter: k - a positive integer that specifies the number of degrees of freedom (i.e. the number of X_i)

The chi-square distribution is a special case of the gamma distribution.



Probability density function of the χ2 distribution



Cumulative distribution function of the χ2 distribution