CLUSTERING INFORMATION ENTITIES BASED ON STATISTICAL METHODS

Von der Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des Grades

Doktor der Naturwissenschaften

Dr. rer. nat.

genehmigte Dissertation von

M. Sc. Marco Fisichella

geboren am 10. September 1982, in Reggio di Calabria, Italien

Hannover, Deutschland, 2012

Referent: Prof. Dr. techn. Wolfgang Nejdl
Ko-Referent: Prof. Dr. Kurt Schneider
Tag der Promotion: 20. Dezember 2012

ZUSAMMENFASSUNG

Mit dem rasanten Wachstum des World Wide Web sind mehr und mehr Informationen digital verfügbar geworden. Die Frequenz, mit der neue Inhalte verfügbar werden, wächst stetig. Bei der Erstellung von Webinhalten existieren nahezu keine zentralen Kontrollmechanismen. Ferner sind die Menschen, die Webinhalte erstellen, sowie die Motive dieser Menschen sehr unterschiedlich. Diese Aspekte erschweren explorative Datenanalysen im Web [MRS08].

Wir konzentrieren uns auf einen wichtigen Schritt von explorativen Datenanalysen: der Clusteranalyse. Die Clusteranalyse beschäftigt sich mit der Klassifizierung von Mustern (Beobachtungen, Datenelemente oder sogenannte Merkmalsvektoren) in Gruppen (Cluster). Die Clusteranalyse ist ein Kernproblem in verschiedenen wissenschaftlichen Disziplinen. Lösungen dieses Problem sind daher von besonderer Wichtigkeit.

In dieser Doktorarbeit, präsentieren wir Lösungen zum Clustern von Informationsentitäten basierend auf statistischen Methoden. Unser Ziel ist es Erkenntnisse mit Bezug auf fundamentale Konzepte zu erzielen, die für eine breite Gruppe von Wissenschaftlern und Praktikern nutzbar sind, die sich mit dem Problem der Clusteranalyse beschäffigen.

Wir beschreiben drei wichtige Anwendungen von Clusteranalyse-Algorithmen im Bereich Information Retrieval: (1) Ähnlichkeitsbasierte Suche nach Datenpunkten hoher Dimensionalität mit dem Ziel Duplikate in Bilderkollektionen (Near Duplicate Images) zu identifizieren, (2) Messung von latenten Variablen in den Sozialwissenschaften um Charakteristiken in wissenschaftlichen Netzwerken zu visualisieren und (3) generative Modelle für die Analyse von Dokumenten, die in natürlicher Sprache verfasst sind, mit dem Zweck der Ereigniserkennung.

Im Rahmen unserer Forschung werden wir typische Aktivitäten der Clusteranalyse erklären wie etwa: (1) Repräsentation von Mustern (insbesondere auch die Extraktion und Selektion von Merkmalen), (2) Definition einer Metrik für die Nachbarschaft von Mustern innerhalb einer Domäne von Daten, (3) Gruppierung von Mustern und (4) Auswertung der Resultate der Clusteranalyse [JMF99].

Die *Repräsentation von Mustern* bezieht sich auf die Anzahl an Klassen/ Gruppen, die Anzahl der verfügbaren Muster sowie die Anzahl, Art und Skalierung von Merkmalen, die einem Clusteranalyse-Algorithmus zur Verfügung gestellt werden. Dies beinhaltet die *Auswahl von Merkmalen*, also den Prozess der Identifikation der effektivsten Merkmale aus einer Menge an grundsätzlich verfügbaren Merkmalen. Ferner umfasst die Repräsentation von Mustern auch

die *Extraktion von Merkmalen*, also den Gebrauch einer oder mehrerer Transformationen der Eingabemerkmale um neue, hervorstechende Merkmale zu generieren. Beide Techniken ermöglichen somit die Erkennung von geeigneten Merkmalen für die Clusteranalyse.

Für die *Nachbarschaft von Mustern* wird typischerweise eine Distanzfunktion angewendet, die die Ähnlichkeit zweier gegebner Muster angibt. In den verschiedenen Forschungsdomänen werden hierzu eine Reihe verschiedener Distanzmetriken eingesetzt, die wir ebenfalls in dieser Arbeit vorstellen.

Die *Gruppierung* kann auf verschiedene Arten durchgeführt werden. Clusteranalyse - Algorithmen gruppieren in diesem Schritt eine initiale Menge von Mustern in eine Teilmenge (Cluster). Hierbei ist das Ziel Cluster zu erstellen, die in sich kohärent sind aber klar unterschiedlich zu anderen Clustern sind. Muster innerhalb eines Clusters sollen demnach so ähnlcih wie möglich sein während ein Muster aus einem Cluster möglichst unterschiedlich zu den Mustern aus den anderen Clustern sein soll. Je größer die Ähnlichkeit innerhalb einer Gruppe und je größer der Unterschied zu anderen Gruppen desto besser das Clustering.

Die *Auswertung der Resultate der Clusteranalyse* betrifft letztlich die Evaluierung des Clusteranalyse - Algorithmen. Diese Auswertung bezieht sich meist auf ein bestimmtes Kriterium für ein Optimum. Allerdings sind diese Kriterien oftmals subjektiv, weshalb wenige sogenannte *Golden Standards* für die Bewertung von Clusteranalyse-Algorithmen existieren. Die eigentlichen Bewertungen sollen objektiv sein [Dub93] und werden durchgeführt um festzustellen, ob die Resultate der Clusteranalyse sinnvoll sind. Im Allgemeinen wird in dieser Phase somit die Struktur des Clustering validiert und analysiert ob das Ergebnis statistisch signifikant ist.

In dieser Doktorarbeit führen wir eine Reihe von Evaluierungen der vorgestellten Methoden durch, die teils auf Benutzerbewertungen und teils auf anderen Datensätzen, die Informationen über die Qualität des Clustering enthalten, basieren. Unsere Evaluierungen zeigen die hohe Qualiyät unserer Lösungen. Abschließend geben wir Einblicke in mögliche Erweiterungen unserer Ansätze und Alternativen für zukünftige Arbeiten.

SCHLAGWORTE
Information Retrieval, Clusteranalyse, statistischen Methoden

ABSTRACT

The booming growth of the World Wide Web has made more and more information available digitally at unprecedented rates and levels of popularity. Also, the Web itself can be considered unprecedented in the almost complete lack of coordination in its creation and in the diversity of backgrounds and motives of its participants. Each of these contributes in making exploratory data analysis hard [MRS08].

In particular, we will focus on one of the steps in exploratory data analysis that is the clustering phase. Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters). The clustering problem has been addressed in many contexts and by researchers in many disciplines; this reflects its broad appeal and usefulness.

In this thesis, we present approaches on Clustering Information Entities based on Statistical Methods, with the goal of providing useful advice and references to fundamental concepts accessible to the broad community of clustering practitioners.

We describe three important applications of clustering algorithms in Information Retrieval: (1) *Similarity Search for High Dimensional Data Points*, with the purpose to find Near Duplicate Images; (2) *Measuring Latent Variable in Social Sciences*, with the aim to visualize Research Communities; and (3) *Generative Model for Content Analysis of Natural Language Documents* to detect Events.

Through our research, we will deepen into typical clustering activities, which involve all or some of the following steps: (1) pattern representation (including feature extraction and/or selection); (2) definition of a pattern proximity measure appropriate to the data domain; (3) clustering/grouping; and (4) assessment of output (if needed) [JMF99].

In detail, *pattern representation* refers to the number of classes, the number of available patterns, as well as the number, the type, and the scale of the features available to the clustering algorithm. This includes *feature selection*, which is the procedure of detecting a subset of the most effective features starting from the original feature set. Furthermore, patter representation comprises *feature extraction*, which applies one or several transformations of the input features to produce new salient features. Either one or both of these techniques can be adopted to obtain a more appropriate set of features for clustering.

Pattern proximity is usually measured by a distance function defined on pairs of patterns. A variety of distance measures are in use in the various

communities and will be described when used in our research.

Grouping step can be performed in a variety of ways. In such a step, clustering algorithms technically group an initial set of patterns into subsets or clusters. The algorithms' goal is to create clusters that are coherent internally, but clearly different from each other. In other words, patterns within a cluster should be as similar (or related) as possible; and patterns in one cluster should be as dissimilar (or unrelated) as possible from patterns in other clusters. The greater the similarity is within a group and greater is the difference between groups, the better and more distinct the clustering.

Finally, *assessment of output* is the evaluation of a clustering procedure's output. Often this assessment uses a specific criterion of optimality; however, these criteria are usually arrived at subjectively. Hence, little in the way of *gold standard* exists in clustering except in well-prescribed sub-domains. Assessments are objective [Dub93] and are performed to determine whether the output is meaningful. In general, this phase is used to validate a clustering structure and to investigate if it cannot reasonably have occurred by chance or as an artifact of a clustering algorithm.

To conclude, we conduct extensive evaluations of the proposed methods against user judgments, as well as against ground truth dataset, revealing the high quality of our approaches. Lastly, we provide insights into possible extensions and future work.

KEYWORDS
Information Retrieval, Clustering, Statistical Methods

FOREWORD

The algorithms presented in this thesis have been published or are under submission at various conferences or journals, as follows.

In Chapter 2, we describe contributions included in:

- Marco Fisichella, Fan Deng, and Wolfgang Nejdl. Efficient incremental near duplicate detection based on locality sensitive hashing. In ***DEXA***: *Proceedings of the International Conference on Database and Expert Systems Applications*, pages 152–166, 2010. [FDN10]
- Marco Fisichella, Fan Deng, and Wolfgang Nejdl. Similarity search for high dimensional data points. In ***TKDE***: *Under Submission at IEEE Transactions on Knowledge and Data Engineering Journal.* [FDN]

Chapter 3 is built upon the work published in:

- Marco Fisichella, Eelco Herder, Ivana Marenzi, and Wolfgang Nejdl. Who are you working with? - visualizing TEL research communities -. In ***ED-MEDIA***: *Proceedings of the International Conference on Educational Multimedia, Hypermedia & Telecommunications*, 2010. [FHMN10]

Finally, in Chapter 4 we include our research presented in:

- Marco Fisichella, Avaré Stewart, Kerstin Denecke, and Wolfgang Nejdl. Unsupervised public health event detection for epidemic intelligence. In ***CIKM***: *Proceedings of the International Conference on Information and Knowledge Management*, pages 1881–1884, 2010. [FSDN10]
- Marco Fisichella, Avaré Stewart, Alfredo Cuzzocrea, and Kerstin Denecke. Detecting health events on the social web to enable epidemic intelligence. In ***SPIRE***: *Proceedings of the International Symposium on String Processing and Information Retrieval*, pages 87–103, 2011. [FSCD11]
- Marco Fisichella and Wolfgang Nejdl. Generative model for content analysis of natural language documents. In ***TKDE***: *Under Submission at IEEE Transactions on Knowledge and Data Engineering Journal.* [FN]

During the stages for my Ph.D. studies, I have also published a number of papers investigating different areas of Information Retrieval. Not all researched areas are touched in this thesis due to space limitation, but the complete list of publications follows:

- Marco Fisichella, Alessandra Pandolfi, and Valerio Targon. Risk government in dangerous goods transportation. In ***AED**: Proceeding of the Advanced Engineering Design Conference*, 2006. [FPT06]

- Marco Fisichella, Alessandra Pandolfi, Valerio Targon, Luciano Raso, and Fabio Siragusa. Dangerous goods governance. In *Multidisciplinarity and innovation, ASP Projects 1, Telesma Edizioni, Lomazzo (Co)*, 2007. [FPT+07]

- Juri Luca De Coi, Marco Fisichella, and Maristella Matera. Managing adaptivity in web collaborative processes using policies and user profiles. In ***ICWE Workshop** on Semantic Web Information Management*, pages 150–162, 2010. [CFM10]

- Katja Niemann, Uta Schwertel, Marco Kalz, Alexander Mikroyannidis, Marco Fisichella, Martin Friedrich, Michele Dicerto, Kyung-Hun Ha, Philipp Holtkamp, and Ricardo Kawase. Skill-based scouting of open management content. In ***EC-TEL**: Proceedings of the European Conference on Technology Enhanced Learning*, pages 632–637, 2010. [NSK+10]

- Avaré Stewart, Marco Fisichella, and Kerstin Denecke. Detecting public health indicators from the web for epidemic intelligence. In ***eHealth**: Proceedings of the International ICST Conference on Electronic Healthcare*, pages 10–17, 2010. [SFD10]

- Kerstin Denecke, Ernesto Diaz-Aviles, Peter Dolog, Tim Eckmanns, Marco Fisichella, Ricardo Gomez-Lage, Jens Linge, Pavel Smrz, and Avaré Stewart. The medical ecosystem [m-eco] project: Personalized event-based surveillance. In ***IMED**: International Meeting on Emerging Diseases and Surveillance*, 2011. [DDAD+11]

- Alfredo Cuzzocrea and Marco Fisichella. A flexible graph-based approach for matching composite semantic web services. In ***EDBT/ICDT Workshop** on Linked Web Data Management*, pages 30–31, 2011. [CF11b]

- Marco Fisichella and Maristella Matera. Process flexibility through customizable activities: A mashup-based approach. In ***ICDE Workshop** on Data Management and Analytics for Semi-Structured Business Processes*, pages 226–231, 2011. [FM11]

- Alfredo Cuzzocrea, Juri Luca De Coi, Marco Fisichella, and Dimitrios Skoutas. Graph-based matching of composite owl-s services. In ***DASFAA Workshop*** *on Graph-structured Data Bases*, pages 28–39, 2011. [CCFS11]

- Ernesto Diaz-Aviles, Marco Fisichella, Ricardo Kawase, Wolfgang Nejdl, and Avaré Stewart. Unsupervised auto-tagging for learning object enrichment (best paper award). In ***EC-TEL****: Proceedings of the European Conference on Technology Enhanced Learning*, pages 83–96, 2011. [DAFK+11]

- Alfredo Cuzzocrea and Marco Fisichella. Discovering semantic web services via advanced graph-based matching. In ***SMC****: Proceedings of the International Conference on Systems, Man and Cybernetics*, pages 608–615, 2011. [CF11a]

- Marco Fisichella and Alfedo Cuzzocrea. Improving flexibility of workflow management systems via a policy-enhanced collaborative framework. In ***WEBIST****: Proceedings of the International Conference on Web Information Systems and Technologies*, 2012. [FC12]

- Marco Fisichella, Ricardo Kawase, Juri Luca De Coi, and Maristella Matera. User profile based activities in flexible processes. In ***WIMS****: Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, 2012. [FKCM12]

ACKNOWLEDGMENTS

Questa tesi costituisce la fine di un percorso ricco di esperienze, bello, ma anche molto duro, che ha costituito il mio dottorato. Senza laiuto di alcune persone, questo percorso sarebbe stato ancora pi arduo per essere calcato. Qui, io mi rivolgo a loro. Voglio ringraziare in primis la mia famiglia, in particolare mio padre, mia mamma e mio fratello. Ovviamente un bacio particolare va a mio nonno. Loro hanno saputo starmi vicino incondizionatamente e sempre. Non potevo ricevere in dono una famiglia migliore. Voglio ringraziare la mia Lucia per tutto lamore che, come un diamante nel cielo, ha saputo darmi condendolo con tanta pazienza. Voglio ringraziare i miei amici, in particolare Ivana, Ricardo e George per le tante piacevoli chiacchierate, il supporto e la leggerezza con cui abbiamo affrontato assieme questo periodo. Voglio ringraziare il Professore Wolfgang per la spontaneit e la concretezza con cui mi ha aiutato. Voglio ringraziare tutte quelle persone che hanno incrociato la mia strada e vedendomi mi hanno sorriso col cuore. Infine, voglio ringraziare me stesso per la caparbiet e la determinazione con cui ho guadagnato centimetro per centimetro questo traguardo.

This thesis is the end of a path full of experiences, beautiful, but also very hard, which was my Ph.D. Without the help of some people, this path would have been even more difficult to be trodden. Here, I thank them. In primis, I want to thank my family, especially my father, my mother, and my brother. Of course, a special kiss goes to my grandfather. They were able to stay close to me unconditionally and always. I could not receive as a gift a better family. I want to thank Lucy, my diamond in the sky, for all the love and the patience she gave me. I want to thank my friends, particularly Ivana, Ricardo and George for the many pleasant chats, support and the ease with which we dealt with this period. I want to thank Professor Wolfgang for his spontaneity and his concreteness, which helped me. I want to thank all those people who crossed my path and who donated me a smile with their hearts. Finally, I want to thank myself for the tenacity and the determination with which I have earned each inch of this goal.

Contents

List of Figures

1 Introduction

Clustering, also called cluster analysis, divides data into groups (clusters) that are meaningful, useful, or both. If meaningful groups are the goal, then the clusters should capture the natural structure of data. Clustering did not begin with the Web. In response to various challenges of providing information access, the field of clustering evolved to give principled approaches to gather various forms of content and more general information. It is difficult to fix a beginning of the field.We could imagine about clustering in biology where the system of nature is categorized through the three kingdoms of nature, according to classes, orders, genera, and species.

Clustering is a main task of explorative data mining, and a common technique for statistical data analysis used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

In recent years, a principal driver of innovation has been the World Wide Web, unleashing publication at the scale of tens of millions of content creators. This explosion of published information would be moot if the information could not be found, annotated, classified, and analyzed so that users can quickly find information that is both relevant and comprehensive for their needs. By the late 1990s, many people felt that continuing to index, classify, categorize, and cluster the whole Web would rapidly become impossible, due to the Web's exponential growth in size. But major scientific innovations, superb engineering, the rapidly declining price of computer hardware, and the rise of a commercial underpinning for web search have all conspired to power todays major search engines and cluster techniques, which are able to provide high-quality results within fractions of second response times [MRS08].

Technically, clustering algorithms group a set of objects into subsets or clusters. The algorithms' goal is to create clusters that are coherent internally, but clearly different from each other. In other words, objects within a cluster should be as similar (or related) as possible; and objects in one cluster should be as dissimilar (or unrelated) as possible from objects in other clusters. The greater the similarity is within a group and greater is the difference between groups, the better and more

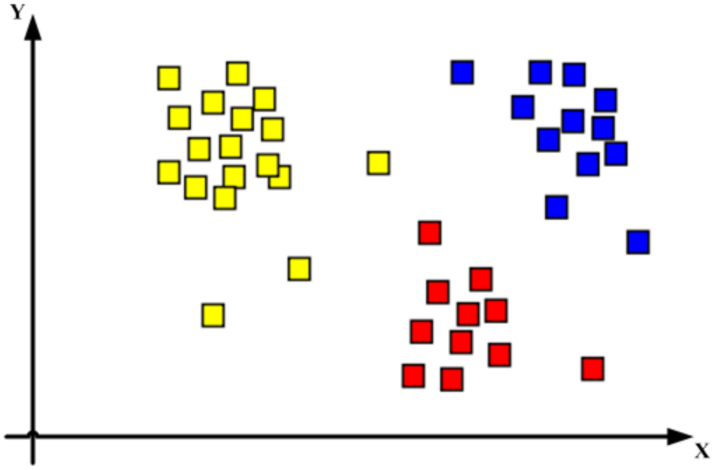


Figure 1.1 An example of a data set with a clear cluster structure [1]

distinct the clustering.

Clustering is the most common form of unsupervised learning. No super-vision means that there is no human expert who has assigned items to classes. In clustering, it is the distribution and make-up of data that will determine cluster membership. A simple example is reported in Figure 1.1. It is visually clear that there are three distinct clusters of points. In this thesis, we propose algorithms that find such clusters in an unsupervised fashion using statistical approaches. Unlike classification, which is a form of supervised learning where the goal is to replicate a categorical distinction that a human supervisor imposes on the data, in unsupervised learning, where clustering is the most important example, we have no such teacher to guide us [MRS08].

In this thesis, we apply *Flat clustering* which creates a flat set of clusters without any explicit structure that would relate clusters to each other. Opposite is the *Hierarchical clustering* which creates a hierarchy of clusters and it is out of the scope of this work.

A second important distinction can be made between *Hard* and *Soft clustering* algorithms. Hard clustering computes a hard assignment: each object is a member of exactly one cluster. The assignment of soft clustering algorithms is soft: the assignment to an object is a distribution over all clusters. In a soft assignment, an object has fractional membership in several clusters. Locality sensitive hashing, Expectation-Maximization (or EM) algorithm, and principal component analysis, a form of dimensionality reduction, can be reduced to *Soft Clustering* algorithms and will be treated in this research.

Also, in many applications, the notion of a cluster is not well defined. The definition of a cluster can be imprecise and it depends on the nature of data and on the desired results. To better understand the difficulty of deciding what constitutes

[1] Image under Creative Commons License avaliable at http://www.wikipedia.org

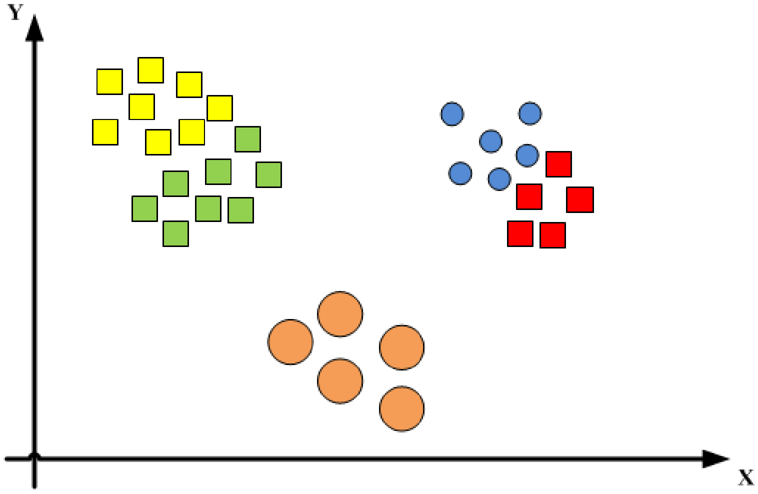


Figure 1.2 An example of a data set with an unclear cluster structure

a cluster, consider Figure 1.2, which shows thirty-two points in two dimensions. The shapes of the objects indicate cluster membership. The apparent division of each of the three larger groups in there clusters may be simply an artifact of the human visual system. Nevertheless, in Figure can be counted five clusters highlighted by different shapes of items. Thus, according to the definition of a cluster, we can have several outcomes. In this work, we provide some specific examples, introducing each time the cluster definition.

Finally, we motivate the use of clustering in information retrieval by introducing a number of applications and define problems we are trying to solve with clustering.

1.1 Clustering in Information Retrieval

1.1.1 Similarity Search for High Dimensional Data Points

Similarity search is an important research topic which finds applications in different areas. For example, finding all similar images of a query image in a large image collection based on certain similarity measures and thresholds. Feature vectors can be extracted from the images. Once this is done, the set of images can be considered as a set of high dimensional points. In general, similarity search can refer to a variety of related problems. The motivating application of this work is online near-duplicate detection for multimedia content sharing websites like Flickr [Fli] and Youtube [You]. Whenever a user is uploading an image or a video, it would be desirable if near-duplicates that are very similar (content-wise) to the one being uploaded can be retrieved and returned to the user in real-time. In this way, the user can identify redundant copies of the object promptly and decide if she should continue the upload. In addition to personal users, enterprise users may also need this type of applications.

For example, media companies such as broadcasters and newspapers may continuously upload their images or videos to a multimedia content repository. The copy-right issue is one of their main concerns. It would be a useful feature if near-duplicate copies can be retrieved and reported to the users during the upload period so that the user can identify pirated copies promptly. If the new object is illegal, the user should immediately stop the upload process.

In summary, the problem we consider is to answer range search queries in an incremental manner. That is, whenever a new point arrives, find all similar/close points (based on a pre-specified similarity threshold) from the set of high dimensional points arrived earlier, and then decide to insert the new point into the data set.

Based on a well-known indexing technique, *Locality Sensitive Hashing* (LSH), we propose a new approach which clearly speeds up the running time of LSH indexing while using only a small amount of extra space. The idea is to store a small fraction of near-duplicate pairs within the existing set which are found when they are inserted into the data set, and use them to prune LSH candidate sets for the newly arrived point.

The basic idea of LSH is to use certain hash functions to map each multi-dimensional point into different clusters, namely buckets, based on their hash values. An essential part of LSH is the hash function family H which uses a peculiarity of the Gaussian distribution. Finally, the nice property of the LSH is to respect the cluster hypothesis to gather together into buckets similar points, proportionally to their similarity.

1.1.2 Measuring Latent Variables in Social Sciences

"In the social sciences, we are often trying to measure things that cannot directly be measured (so–called latent variables)", as Andy Field states in his book [Fie09]. The interest in different topics or research areas of different authors within the scientific community cannot easily be measured. We could not measure motivation and interest directly, but we tried to analyze a possible underlying variable (collaboration in the form of co-citations among the major authors), to detect different sub-communities and possible trends. We tried to answer to the questions: "what communities and sub-communities can be identified in technology enhanced learning (TEL) area", "what research topics/specialties can be identified in a field of studies", and "what conferences are the most relevant for what topic and for what community".

Being aware of this fragmentation and of the various sub-communities which make up the TEL area is an important pre-requisite towards overcoming this fragmentation, increasing synergies between different sub-areas and researchers, and, last but not least, providing funding agencies with evidence of new research results, innovative applications and promising new approaches for technology-enhanced learning.

Author Co-Citation Analysis (ACA) provides a principled way of analyzing research communities, based on how often authors are cited together in scientific publications. In this work, we present results based on ACA to analyze and visualize research communities in the area of technology-enhanced learning, focusing on publicly available citations and conference information. We describe our approach to collecting, organizing and analyzing appropriate data, as well as the problems which have to be solved in this process.

To do so, we used the statistical Principal Component Analysis (PCA) to detect appropriate clusters in TEL research, and then visualize and interpret these clusters. Specifically, PCA is a technique for identifying groups or clusters of variables and for reducing the data set to a more manageable size while retaining as much of the original information as possible. Often, its operation can be thought of as revealing the internal structure of the data in a way which best explains the variance in the data.

Finally, we also provide a thorough interpretation of the obtained TEL research clusters, which offer insights into these research communities.

1.1.3 Generative Model for Content Analysis of Natural Language Documents

Content analysis and clustering of natural language documents/articles become crucial in various domains. Clustering documents serves to extract events, where an event is defined as a specific thing happening at a specific time and place, which may be consecutively reported by many articles in a period under observation.

In this thesis, we introduce an approach for clustering articles in an unsupervised manner. Unsupervised learning means no supervision, thus there is no human expert who assigned documents to classes. Clustering is the most common form of unsupervised learning.

Also, we have chosen a probabilistic generative model for event detection, because it has been proven to be a more unified framework for handling the multiple modalities (i.e. time, content, and entity types) of a document and its content. The generative modeling approach to information retrieval directly models the idea: a document is a good match to an event definition if the document model is likely to generate the event definition, which will in turn happen if the document contains often the keywords defining the event.

Through our research, we will present an application scenario of our methodology within the Retrospective Event Detection area (RED) that is part of the clustering area algorithms which have the task to discover previously unidentified events in a historical collection. The specific and important domain of health boosted us to apply our approach to real needs in the medical area. In particular, the detected events we extract are defined as Public Health Events (PHE). Actually, a PHE is intended to be some emerging infection, symptom, or illness affecting people or animals in a particular geographic place during a specific time period.

We prove that applying an unsupervised algorithm to public health event detection will help epidemiologists to mitigate the impact of potential diseases-spreading detecting the medical event as early as possible.

1.2 Contributions of this Thesis

Our various contributions to Clustering in IR are summarized as follows:

- We provide a solution to the problem of detecting near-duplicates for high dimensional points in an incremental manner.
- We develop an approach to create clusters of near-duplicate images, considered as high dimensional points.
- We offer a principled way of analyzing research communities, based on authors cited together in scientific publications.
- We ease authors' work to find collaboration between researchers within the same scientific community, and also increasing synergies between different sub-communities and researchers.
- We propose an approach for unsupervised event detection and we adapt that to the domain of public health event detection (medical area).
- We help epidemiologists to mitigate the impact of potential diseases-spreading detecting the medical event as early as possible.

1.3 Thesis Structure

We start in Chapter 2 presenting similarity search for high dimensional data points problem. Since our approach is based on a well known indexing technique such as Locality Sensitive Hashing (LSH), we first present some background knowledge of LSH and related works in Section 2.2; we then introduce our SimPair LSH approach and show the algorithm analysis in Section 2.3; also in this section, we propose an algorithm predicting the gain of our new approach and provide some analysis. In Section 2.4, we demonstrate the superiority of our approach over LSH via extensive experiment results. Last, we give a conclusion over the problem and the approach to solve that in Section 2.5.

In Chapter 3, we cope with measuring latent variables in social sciences. We use author co-citation analysis supported by principal component analysis (PCA) to analyze and visualize research communities in the area of technology-enhanced learning, focusing on publicly available citation information provided through CiteseerX

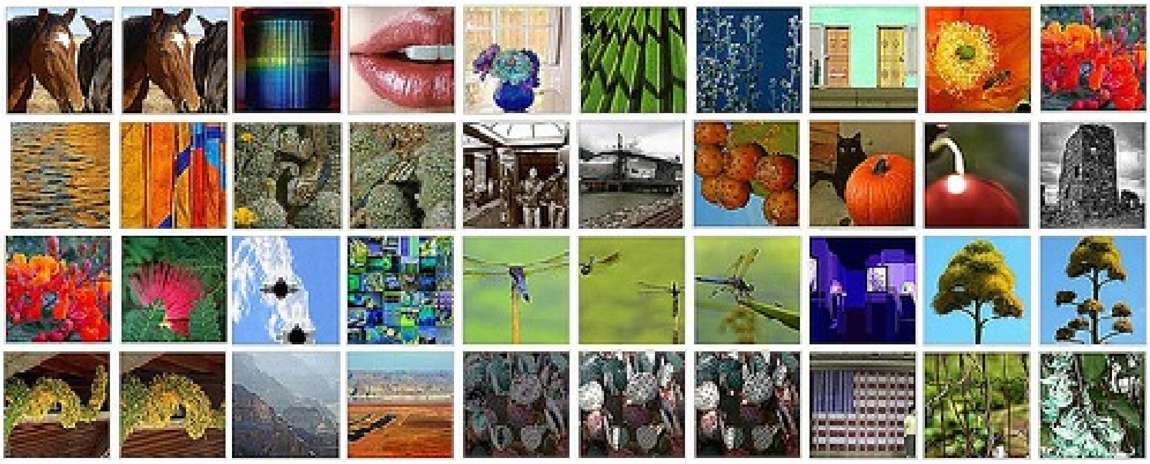
[Cit] and conference information available through DBLP [DBLa]. In Section 3.2, we present some background knowledge on PCA and related works. Thus, we introduce our approach in Section 3.3. Furthermore, in Section 3.4 the results, in term of research communities identified, are visualized based on relevant conferences and themes for each cluster, providing a first important step to a structured overview over researches in technology-enhanced learning and make TEL researchers aware of the different research communities relevant for their work. Finally, we summarize and conclude our results in Section 3.5.

Our approach with generative model for content analysis of natural language documents is presented in Chapter 4. This chapter is organized as follows: we discuss related works and retrospective event detection in Section 4.2. In Section 4.3, we present details of our generic approach based on a generative model for clustering documents, while in Section 4.4, we characterize the nature of event detection in the public health domain, namely the medical domain, to lay the foundation for describing the task-specific adaptations required in this setting. We provide experimental results for our approach in Section 4.5. Finally, in Section 4.6, we conclude and outline future work in this area.

Finally, we conclude the thesis in Chapter 5 with an enumeration of contributions we brought to Clustering Information Entities based on Statistical Methods in Information Retrieval research, while giving an overview as well on possible future research directions and open challenges associated to these topics.

2

Efficient Similar Pair Information Maintenance based on LSH



[1]

In this chapter, we study the problem of detecting near-duplicates for high dimensional data points in an incremental manner. For example, for an image sharing website, it would be a desirable feature if near-duplicates can be detected whenever a user uploads a new image into the website so that the user can take some action such as stopping the upload or reporting an illegal copy. Specifically, whenever a new point arrives, our goal is to find all points within an existing point set that are close to the new point based on a given distance function and a distance threshold before the new point is inserted into the data set. Based on a well-known indexing technique such as Locality Sensitive Hashing, we propose a new approach which clearly speeds up the running time of LSH indexing while using only a small amount of extra space. The idea is to store a small fraction of near-duplicate pairs within the existing point set which are found when they are inserted into the data set, and use them to prune LSH candidate sets for the newly arrived point. Extensive experiments based on three real-world data sets show that our method consistently outperforms the original LSH approach: to reach the same query response time, our method needs significantly less

[1] Image under Creative Commons License avaliable at http://www.flickr.com/photos/cobalt/552941780/sizes/l/in/photostream/

memory than the original LSH approach. Meanwhile, the LSH theoretical guarantee on the quality of the search result is preserved by our approach. Furthermore, our approach based on LSH is easy to implement.

2.1 Introduction

Similarity search is an important research topic which finds applications in different areas. For example, finding all similar images of a query-image in a large image collection based on certain similarity measures and thresholds. Feature vectors can be extracted from the images. Once this is done, the set of images can be considered as a set of high dimensional points. In general, similarity search can refer to a variety of related problems. In this work, the problem we consider is to answer range search queries in an incremental manner. That is, whenever a new point arrives, find all similar/close points (based on a pre-specified similarity threshold) from the set of high dimensional points arrived earlier, and then insert the new point into the data set.

The motivating application of this work is online near-duplicate detection for multimedia content sharing websites like Flickr [Fli] and Youtube [You]. Whenever a user is uploading an image or a video, it would be desirable if near-duplicates that are very similar (content-wise) to the one being uploaded can be retrieved and returned to the user in real-time. In this way, the user can identify redundant copies of the object promptly and decide if she should continue the upload. In addition to personal users, enterprise users may also need this type of applications. For example, media companies such as broadcasters and newspapers may continuously upload their images or videos to a multimedia content repository. The copy-right issue is one of their main concerns. It would be a useful feature if near-duplicate copies can be retrieved and reported to the users during the upload period so that the user can identify pirated copies promptly. If the new object is illegal, the user should immediately stop the upload process.

Compared to the traditional similarity search problem, fast response is more important for this type of applications since similarity search is only part of the online content upload process which must be completed within a few seconds at most. In addition to the online requirement, another characteristic of the motivating applications is that the similarity search operations are executed together with data point insertions. In other words, the data set is created incrementally where the near neighbors of each point are known before the point is inserted into the data set.

To speed up the searching process, in-memory indexing techniques are ideal solutions if the help of disk-based index are not necessary since a disk access is an order of magnitude slower than a memory operation. For a data set with 1 million points, an index storing all the point IDs once only needs 12MB memory assuming that each ID takes 12 bytes; if each point is a 162-dimensional point and each dimension of a point takes 4 bytes, storing all the points requires 648MB, which is tolerable even for an inexpensive PC nowadays. Although processing Web-scale data set with billions of points may need clusters with tens or hundreds of distributed machines, indexing an enterprise-scale data set with tens or hundreds of millions points in main-memory is feasible using a single server with a larger memory size. Unfortunately, to give a fast query response, the index size needed for high-dimensional points is usually larger than the size we computed, and it can be even larger than the data set size. Thus, in this work we focus on reducing memory consumption of in-memory index while providing fast query response.

Although decades of research have been conducted on similarity search, the problem is still considered challenging. One important reason is the "curse of dimensionality". It has been shown that exponential space in n (number of points in the data set) is needed to speed up the similarity search process; in other words, the searching time increases exponentially with the dimensionality [AI08]. It has also been shown both theoretically and empirically [WSB98] that all partitioning and clustering based indexing approaches degrade to a brute force linear scan approach when the dimensionality is sufficiently high.

To our knowledge, a state-of-the-art solution to the similarity search problem in practice, which provides fast query response time, is the Locality Sensitive Hashing (LSH)[IM98] although it has been proposed for a decade. Meanwhile, LSH also provides theoretical guarantees on the quality of the solution. However, also suffering from the "curse of dimensionality", LSH needs large amount of space to achieve fast query response.

In the rest of this chapter, we first present some background knowledge of LSH and related work; we then introduce our SimPair LSH approach and show the algorithm analysis; also in this section, we propose an algorithm predicting the gain of our new approach and provide some analysis. In Section 2.4, we demonstrate the superiority of our approach over LSH via extensive experiment results. Last, we conclude in Section 2.5.

2.1.1 Problem statement (Incremental Range Search)

In this study, we focus on the incremental range search problem defined as follows: Given a point q and a set P with n d-dimensional points, efficiently find out all points in P that are similar to q based on certain similarity/distance function and a similarity threshold τ before q is inserted into the data set. We call the points similar to q *near neighbors* of q. In this problem, before evaluating the query q, the near neighbors of all points within the data set are retrieved when they are inserted into the data set.

Distance measure. We focus on Euclidean distance since it has been widely used in different applications. It is not hard to extend the technique to other distance functions such as L1 and Hamming distance, as the underlying technique, Locality

Sensitive Hashing, can be applied in those cases.

In-memory index structure. We focus on in-memory index structure since fast real-time response is the first priority in the applications we consider. For high dimensional similarity search, the index size can be as large as, or even larger, than the data set size in order to give an efficient query response time. Therefore, reducing the memory cost while providing fast response is the main concern of this work.

2.1.2 Straightforward solution

A straightforward solution to this problem is LinearScan: compute the distance between q and each point p in P; if the distance is above the given similarity threshold, output this point. It is not hard to see that this approach can be very slow for large data sets, especially when the dimensionality d is large; in the case of Euclidean distance, LinearScan takes $O(nd)$ time for each query.

2.1.3 Our contributions

The contributions of this work are:

- We proposed a novel approach, namely SimPair LSH, to speed up the original LSH method; the main idea is to take advantage of a certain number of existing similar point pairs and use them to prune LSH candidate sets relevant for a given query.
- The correctness and effectiveness of the new approach is analyzed; an algorithm predicting the gain and SimPair LSH is provided and verified by our experiments; and details maintaining the similar pairs are described.
- Thorough experiments conducted on 3 real-world data sets show that our method consistently outperforms LSH in terms of query time in all cases that we tried, with a small amount of extra memory cost. To achieve the same query time saving, we show that LSH needs significantly more space. Meanwhile, we show that our method preserves the important theoretical guarantee on the recall of query answers.

2.2 Related Work

2.2.1 Locality Sensitive Hashing (LSH)

Locality Sensitive Hashing (LSH) [GIM99, IM98] was proposed by Indyk and Motwani and finds applications in different areas including multimedia near duplicate detection (e.g., [CPIZ07]). LSH was first applied in indexing high-dimensional points for Hamming distance [GIM99], and later extended to L_p distance [DIIM04] where L_2 is Euclidean distance, which we will use in this research.

The basic idea of LSH is to use certain hash functions to map each multi-dimensional point into a scalar; the hash functions used have the property that similar points have higher probability to be mapped together than dissimilar points. When LSH is used for indexing a set of points to speed up similarity search, the procedure is as follows: first, select k hash functions uniformly at random from a LSH hash function family, hereafter introduced, and create L hash tables (buckets); create an index (a hash table) by hashing all points in the data set P into different buckets based on their hash values; when the query point q arrives, use the same set of hash functions to map q into L buckets, one from each hash table; retrieve all points from the L buckets into a candidate set C and remove duplicate points in C; for each point in C compute its distance to q and output those points similar to q.

An essential part of LSH is the hash function family H based on the most well known example of a 2-stable distribution: the *Gaussian distribution*. Generally, stable distributions [Zol86] are defined as limits of normalized sums of independent and identically distributed (i.i.d.) random variables. A distribution D over $\Re$ is called *p-stable*, if there exists $p \geq 0$ such that for any n real numbers $v_1 \ldots v_n$ and i.i.d. variables $X_1 \ldots X_n$ with distribution D, the random variable $\sum_i v_i X_i$ has the same distribution as the variable $(\sum_i |v_i|^p)^{1/p} X$, where X is a random variable with distribution D. The *Gaussian (normal) distribution* is 2-stable. In computer science, stable distributions were used for "sketching" of high dimensional vectors [Ind06] and have found use in various applications. The main property of p-stable distributions mentioned in the definition above directly translates into a sketching technique for high dimensional vectors. The idea is to generate a random vector a of dimension d whose each entry is chosen independently from a p-stable distribution. Given a vector v of dimension d, the dot product $a \cdot v$ is a random variable which is distributed as $(\sum_i |v_i|^p)^{1/p} X$ (i.e. $||v||_p X$), where X is a random variable with p-stable distribution. A small collection of such dot products $(a \cdot v)$, corresponding to different a's, is termed as the sketch of the vector v and can be used to estimate $||v||_p$ (see [Ind06] for details). It is easy to see that such a sketch is linearly composable, i.e. $a \cdot (v_1 v_2) = a \cdot v_1 a \cdot v_2$.

In this work, we will use Gaussian distributions, then $p = 2$. Furthermore, in the approach we will present, such a distribution is used in a slightly different manner as so far described. Instead of using the dot products $(a \cdot v)$ to estimate the $l_2 norm$, we apply them to assign a hash value to each vector v. Intuitively, for Euclidean Distance, i.e. the $l_2 norm$, the hash function family can be constructed as follows [DIIM04]: map a multi-dimensional point p into a scalar by using the function $h(p) = \lfloor \frac{a \cdot p + b}{r} \rfloor$ where a is a random vector whose coordinates are picked uniformly at random from a normal distribution, and b is a random variable uniformly distributed in the range $[0, r]$. For sake of clarity, $h(p)$ constitutes the form of hash functions which belong to the hash function family H. In this hash function, the dot product $a \cdot p$ is projecting each multi-dimensional point p into a random line; the line is cut into multiple intervals with length r; the hash value shows which interval p is mapped to after a random shift of length b. Intuitively, it is clear that closer points have higher chance being mapped into the same interval than distant points under this random projection. Last, generate a new hash function $g(p)$ to be used in constructing a hash table by concatenating k $h_i(p)$ $(i = 1, \dots, k)$, each chosen uniformly at random from the hash function family H, i.e. $g(p) = (h_1(p), \dots, h_k(p))$.

The nice property of the LSH is that the probability that two points p_1 and p_2 are hashed into the same bucket is proportional to their distance c, and this probability can be explicitly computed using the following formulas:

$$p(c) = Pr[h(p_1) = h(p_2)] = \int_0^r \frac{1}{c} f(\frac{t}{c})(1 - \frac{t}{r}))dt, \tag{2.1}$$

where $f(t)$ is the probability density function of the absolute value of the normal distribution. Having $p(c)$, we can further compute the collision probability under H:

$$P(c) = Pr[H(p_1) = H(p_2)] = 1 - (1 - p(c)^k)^L. \tag{2.2}$$

2.2.2 Other LSH-based approaches

Since proposed, LSH has been extended in different directions. Lv et al. [LJW+07] proposed multi-probe LSH and showed experimentally that their method significantly reduced space cost while achieving the same search quality and similar time efficiency compared with original LSH. The key idea of multi-probe LSH is that the algorithm not only searches for the near neighbors in the buckets to which the query point q is hashed, it also searches the buckets where the near neighbors have slightly less chance to appear. The benefit of multi-probe LSH is that each hash table can be better utilized since more than one bucket of a hash table is checked, which decreases the number of hash tables. However, multi-probe LSH does not provide the important search quality guarantee as LSH does. The original LSH scheme guarantees that the true results will be returned by the search algorithm with high probability, while multi-probe could not. This makes multi-probe LSH not applicable in those applications where the quality of the retrieval results are required to be guaranteed. The idea of multi-probe LSH was inspired by earlier work investigating entropy-based LSH [Pan06]. The key idea is to guess which buckets the near neighbors of q are likely to appear in, by randomly generating some "probing" near neighbors and checking their hash values. Similar to multi-probe LSH, entropy-based LSH also reduces the number of hash tables required, though it is difficult to generate proper "probing" near neighbors in a data-independent way [LJW+07].

Another extension of LSH is LSH forest [BCG05] where multiple hash tables with different parameter settings are constructed such that different queries can be handled with different settings. In the theory community, a near-optimal LSH [AIP06] has been proposed; however, currently it is mostly of theoretical interest because the asymptotic running time improvement is achieved only for a very large number of input points [AI08].

Furthermore, a recent work [HMA09], inspired by the idea of Locality Sensitive Hashing (LSH) technique, is the distributed similarity search and range query processing in high dimensional data. Authors consider mappings from the multi-dimensional LSH bucket space to the linearly ordered set of peers that jointly maintain the indexed data and derive requirements to achieve high quality search results and limit the number of network accesses. Locality preserving properties of proposed mappings is proved.

More LSH related work can be found in a recent survey [AI08]. This survey also observes, that despite decades of research, current solutions still suffer from the "curse of dimensionality", i.e. either space or query time exponential effort in the dimensionality d is needed to guarantee an accurate result. In fact, for a large enough dimensionality, current solutions provide little improvement over LinearScan, both in theory and in practice [AI08]. We further note that our technique is orthogonal to other LSH variants described above and, more important, it can be applied in those scenarios.

2.2.3 Tree-based indexing techniques

When the dimensionality is relatively low (e.g., 10 or 20), tree-based indexing techniques are known to be efficient. Examples include kd-trees [Ben75], R-tree[Gut84], SR-tree[KS97], cover-trees [BKL06], and navigating-nets [KL04]. These methods do not scale well with the (intrinsic) dimensionality. Weber et al. [WSB98] show that when the dimensionality exceeds 10, all space partitioning and clustering based indexing techniques degrade to LinearScan. For indexing high dimensional points, B+ tree is also used together with different techniques handling the "dimensionality curse", such as iDistance [YOTJ01] and LDC [KOST04]. Other tree-based approaches like IQ-tree[BBJ$^+$00] and A-tree [SYUK00] use a smaller vector to represent the data points approximately which helps to reduce the complexity of the problem. Different from the LSH based approaches where large amount of space is traded for gaining fast response time, the tree-based approaches have less concern on index space while they usually have faster but comparable query time as LinearScan.

A recent work based by [TYSK10] tried to combine B-trees and LSH. Authors proposed an access method called the Locality-Sensitive B-tree (LSB-tree) to enable fast, accurate, high-dimensional near neighbors search in relational databases. The combination of several LSB-trees forms a LSB-forest that has strong quality guarantees, but improves dramatically the efficiency of the previous LSH implementation having the same guarantees. In practice, the LSB-tree itself is also an effective index which consumes linear space, supports efficient updates, and provides accurate query results. We recall that the technique in this study is orthogonal to the aforementioned LSH variants and can be applied in this scenario.

Due to the intensive research within the past decades, there is a large body of related literature which cannot be covered here. Samet's book [Sam06] provides a comprehensive survey on this topic.

2.2.4 Similarity search and duplicate detection on streaming data

The applications we considered have certain data stream characteristics such as continuous queries and real-time response requirement, although they are still mainly traditional Web applications. Within the past decade, data streaming processing has been a popular topic where the applications include sensor data processing, real-time financial data analysis, Internet traffic monitoring and so on. Gao et al. [GW05] and Lian et al. [L08] studied the problem of efficiently finding similar time series on streaming data, and they achieved efficiency by accurately predicting future data. Their methods are for time series data and cannot be used for the type of applications we consider. Koudas et al. [KOT04] studied the problem of finding k nearest neighbors over streaming data, but they were concerned about low-dimensional case. Deng and Rafiei [DR06] studied the problem of detecting duplicates for streaming data, where no similarity is involved.

2.3 SimPair LSH

Our approach is based on the standard LSH indexing and takes advantage of existing similar pair information to accelerate the running time of LSH. Thus, we call it *SimPair LSH*. Unless noted otherwise, LSH denotes the original LSH indexing method in the rest of this chapter.

2.3.1 Key idea

We observe that LSH retrieves all points stored in the buckets where a query point q was hashed. Let the set of points returned by LSH be the candidate set C. Then, q is compared with all the points in C as in LinearScan, and the near neighbors are found. To guarantee a low chance of missing a near neighbor in C, a large number of hash tables has to be created which may lead to a large C depending on the query q, and accordingly increases the running time especially when d is large.

The main idea of this work is to take advantage of a certain number of pair-wise similar points in the data set and store them in memory; in the process of scanning through C, the search algorithm can look up the similar pair list on-the-fly whenever a distance computation between q and a point p in C is done; if a similar pair $(p, p\prime)$ is found in the list, it is very likely that $p\prime$ will also appear in C; based on the known distances $d(q, p)$ and $d(p, p\prime)$ we can infer $d(q, p\prime)$ by using triangle inequality and may skip the distance computation between q and $p\prime$. The reason why this idea works is that LSH tends to group similar objects into the candidate set C. Thus the points in C are very likely to be similar to each other. Checking one point p can avoid computing distance for the points similar to p, and therefore saving distance computations.

To help readers follow the study, we summarize the symbols used hereafter in Table 2.1.

Table 2.1 The Symbol List

Symbols	Meanings
P	The set of input data points
n	Number of points in the data set
d	Dimensionality of the input points
$d(\cdot,\cdot)$	Distance function
SP	The set of similar point pairs pre-computed
C	Candidate set returned by LSH
q	Query point
p	A point in the candidate set C
$p\prime$	A point in SP
L	Number of hash tables used
k	Number of “small” hash functions generating the real hash function being used
τ	Similarity threshold for the similarity search
θ	Similarity threshold for the similar point pairs stored in SP

2.3.2 The SimPair LSH Algorithm

Our SimPair LSH algorithm works as follows: given a set of points P and all point pairs (including their distances) whose pair-wise distances are smaller than a threshold θ (let the set of all similar pairs be SP). Also given the distance threshold τ determining near neighbors, SimPair LSH then creates L indices as in LSH; whenever a query point q comes, SimPair LSH retrieves all points in the buckets to which q is hashed. Let this set of points be the candidate set C. Instead of scanning through all the points p in C one by one and compute their distances to q as in LSH, SimPair LSH checks the pre-computed similar pair set SP whenever a distance computation $d(q, p)$ is done. Based on the distance between p and q, SimPair LSH continues in 2 different ways:

- If $d(q, p) <= \tau$, SimPair LSH searches in SP for all points $p\prime$ which satisfies $d(p, p\prime) <= \tau - d(q, p)$; check if $p\prime$ in the candidate set C or not; if yes, then mark $p\prime$ as a near neighbor of q without the distance computation.

- If $d(q, p) > \tau$, SimPair LSH searches in SP for all those points $p\prime$ which satisfy $d(p, p\prime) < d(q, p) - \tau$; check if $p\prime$ in the candidate set C or not; if yes, then remove $p\prime$ from C without the distance computation.

The detailed description is shown in Algorithm 1:

```
Algorithm 1: SimPair LSH
Input: A set $P$ with $n$ $d$-dimensional points; $L$ in-memory hash tables created
       by LSH; a set $SP$ storing all similar pairs in $P$ whose pair-wise
       distances are smaller than $\theta$; a distance threshold $\tau$ defining near
       neighbors; and a query point $q$
Output: all near neighbors of $q$ in $P$
begin
    check the $L$ buckets $q$ hashed to and retrieve all the points in those buckets
    as in LSH;
    put all the points into a candidate set $C$;
    for each point $p$ in $C$ do
        compute the distance between $q$ and $p$, i.e. $d(q, p)$;
        if $d(q, p) < \tau$ then
            output $p$ as a near neighbor of $q$;
            search in $SP$ for all the points $p\prime$ which satisfies $d(p, p\prime) < \tau - d(q, p)$;
            for each point $p\prime$ found in $SP$ do
                check if $p\prime$ in $C$ or not;
                if found then
                    output $p\prime$ as a near neighbor of $q$ and remove it from $C$;
        if $d(q, p) > \tau$ then
            search in $SP$ for all the points $p\prime$ which satisfies $d(p, p\prime) < d(q, p) - \tau$;
            for each point $p\prime$ found in $SP$ do
                check if $p\prime$ in $C$ or not;
                if found then
                    remove $p\prime$ from $C$;
end
```

The algorithm constructing the LSH indices is the original LSH algorithm, as described in Algorithm 2.

[DIIM04] describes how to select L and g_i to guarantee the success probability.

Algorithm 2: Constructing LSH indices

```
Input: A set $P$ with $n$ $d$-dimensional points; a distance threshold $\tau$ defining
       near neighbors; $L$ LSH functions $g_1, \dots, g_L$; a success probability
       guaranteeing the chance of including all near neighbors in the result set
Output: $L$ hash tables
begin
    initialize the $L$ hash tables;
    for each point $p$ in $P$ do
        for $i$=1, ..., $L$ do
            store the ID of $p$ in bucket $g_i(p)$
end
```

2.3.3 Algorithm Correctness

Since our algorithm is based on LSH, it is important that the theoretical guarantee still holds for SimPair LSH.

Theorem 1 *SimPair LSH has the same theoretical guarantee as LSH has in terms of the range search problem we study. That is, near neighbors will be returned by SimPair LSH with a user-specified probability by adjusting the parameters (hash functions k and number of hash tables L) accordingly.*

Proof 1 *Since we consider points in metric space where triangle inequality holds, SimPair LSH guarantees that the points skipped are either true near neighbors or not near neighbors without distance computation.*

2.3.4 Algorithm effectiveness

The benefit of SimPair LSH compared with LSH is that points in the candidate set returned by LSH can be pruned by checking the similar pair list SP without distance computations. Therefore, it is important to analyze the number of prunes SimPair generates. Also, to obtain the benefit, SimPair LSH has to search in SP and C for the points to be pruned, which can take time, although hash indices can be built to speed up each search operation to $O(1)$ time. Next, we analyze the factors affecting the gain and cost.

Pruning analysis. To generate a prune from a point p in C, SimPair first has to find a "close enough" point $p\prime$ of p from SP, where close enough or not depends on $|d(q,p) - \tau|$. If $|d(q,p) - \tau|$ is large, SimPair LSH has a higher chance to find a $p\prime$.

Another factor that can affect the chance of finding $p\prime$ from SP is the size of SP. Clearly, maintaining a large set of SP will increase the chance of finding $p\prime$ of p.

Finding $p\prime$ of p does not necessarily lead to a prune. The condition that a prune occurs is that $p\prime$ appears in C. According to the property of LSH hash functions, points close to q have higher chance appearing in C. In other words, $d(q, p\prime)$ determines the chance of generating a prune. Although $d(q, p\prime)$ cannot be precisely known, a bound of this distance can be derived from $d(q, p)$ and the "close enough" threshold $|d(q, p) - \tau|$.

Cost analysis. To gain the pruning, SimPair LSH has to pay certain amount of costs including time and space costs. The time cost mainly comes from the searching processes: find the points "close enough" to p in SP and check those points to see if they are in C or not. By constructing hash indices for SP, searching for p in SP only takes $O(1)$ time; constructing hash indices for SP also takes $O(1)$ time for each object. When a candidate set C of points for the query q is retrieved, all points in the dataset belonging to C are marked both in LSH and SimPair LSH; this is possible since each point in the data set has a Boolean attribute showing if the point is in C or not. The purpose of having this attribute is to remove duplicate points when generating C. Duplicates can appear in C because one point can appear in multiple LSH hash buckets. Note that when the searching is finished, the boolean attributes need to be cleared (for both LSH and SimPair LSH) which takes $O(|C|)$ time when all points in C are also maintained in a linked list. For the sake of pruning, another boolean attribute is needed for each point to indicate if the point has been pruned or not.

With these boolean attributes, searching for $p\prime$ in C takes $O(1)$ time. The time cost is mainly generated by searching $p\prime$ in C since there can be multiple $p\prime$ for each p, and therefore multiple look-ups in C.

In addition to the time cost, SimPair LSH also has some extra space cost for storing SP compared with LSH. This cost is limited by the available memory. In our approach, we always limit the size of SP based on two constraints: (i) the similarity threshold θ (for the similar point pairs stored in SP) is restricted to the range $(0, \tau]$; (ii) the size of SP must not exceed a constant fraction of the index size (e.g., 10%).

2.3.5 Similar pair maintenance

Since the gain from SimPair LSH lies on the fact that similar pair information SP is stored in memory, it is important to maintain that properly.

Data structure. SP can be implemented as a two dimensional linked list. The first dimension is a list of points; the near neighbors of each point q in the first-dimension list are stored in another linked list (the second dimension), ordered by the distances to q. On top of the first-dimension linked list, a hash index is build to speed up the look-up operations.

Bounding the size of SP. Since the total number of all similar pairs for a dataset of n objects can be $O(n^2)$, an underlying issue is how to restrict the size of SP. To achieve this purpose we set θ (the similarity threshold for the similar point pairs stored in SP) to τ (the similarity threshold for the similarity search). However, if the dataset size is large, SP can be too big to fit in memory, thus we impose a second bound on the size of SP: it must not go over a constant fraction of the index size (e.g., 10%). To satisfy the latter constraint we can reduce the value of θ within the range $(0, \tau]$; clearly, a bigger value of θ will generate a larger set of SP increasing the chance of finding $p\prime$ of p in C, and thus triggering a prune.

Another possible solution is to remove certain number of similar pairs with the largest distances when the size of SP is above the space bound. We first estimate the number of similar pairs (say k pairs) to be removed based on the space to be released. The k pairs to be removed should be those whose distances are the largest since they have less chance to generate a prune. Thus, we should find the top-k pairs with the largest distances. To obtain this top-k list, we can create a sorted linked list with k entries, call L_k; we then take the part with the largest distances of the first second-dimension linked list in SP and fill L_k (assuming L_k is shorter than the first second-dimension linked list; if not, we go to the next second-dimension linked list). Then, we scan the second second-dimension linked list and update L_k. After scanning all the second-dimension linked lists, the top-k pairs with the largest distances are found, and we can remove them from SP. Note that real-time updates are not necessary for these operations, and they can be buffered and executed when the data arrival rate is slower. Details will be discussed later.

Inserting a point. In a continuous query scenario, each point (say q) issues a query before inserted into the database. That is, as an application requirement, all near neighbors of each newly arrived point need to be found before the new point updates SP. Hence, to maintain the similar pair list, we only needs to add the near neighbors of q just found into SP and there is no similarity search involved.

To add near neighbors of q into SP, we first sort the near neighbors based on their distances to q, and store these near neighbors in a linked list; we then insert q and the linked list into SP, meanwhile update the hash index of SP. Besides, we need to search for each near neighbor of q and insert q into the corresponding linked lists of its near neighbors. Within the linked list into which q is to be inserted, we use a linear search to find the right place q should be put based on the distance between q and the near neighbor. Note that we could have build indices for the second-dimension linked lists, but this will increase the space cost. Also, real-time update of SP are not necessary unlike the query response; we can buffer the new similar pairs and insert them into SP when the data arrival rate is lower.

Deleting a point. When a point q is to be deleted from the data set, we need to update SP. First, we search for all the near neighbors of q in SP and remove q from each of the second-dimension linked lists of the near neighbors; we then need to remove the second dimension linked list of q from SP.

Buffering the SP updates. As mentioned earlier, the updates to SP (inserting a point and size reducing) can be buffered and processed later when the processor is overloaded since data arrival rate can be bursty: there can be a large number of queries at one time and very few queries at another time. This buffering mechanism is to guarantee the real-time response to the similarity search query. Note that the operation of deleting a point cannot be buffered since this can lead to inconsistent results.

Creating SP for a static data set. In the case that data are static and not created by the incremental process, one can build the similar pairs offline before query arrives using some existing similarity join algorithms (e.g., [JS08]). Since it is out of the scope of this work, we will not discuss this in detail.

2.3.6 Pruning prediction

Although SimPair LSH always outperforms the original LSH in all of our experiments, the amount of improvement differs. It would be useful to provide an option for the practitioners to predict the number of prunes in advance.

The idea. According to the pruning analysis, a lower bound of the number of prunes given a query q can be estimated. The idea is as follows: take a few sample points p from C and for each p compute $d(q, p)$. Based on the sample, estimate the distribution of different $d(q, p)$ for all p in C. From $d(q, p)$ we can derive an upper bound of $d(q, p\prime)$ according to the triangle inequality. Thus, we can estimate a lower bound of the probability that $p\prime$ appears in C, and accordingly a lower bound of the number of prunes.

Again, by using the small fraction of sample points obtained from C, SimPair LSH can check SP and find if there is any “close enough” point $p\prime$ of p such that $d(p\prime, p) < |d(q, p) - \tau|$. Since the distance $d(q, p)$ is known, an upper bound of $d(q, p\prime)$ can be derived according to the triangle inequality. Knowing $d(q, p\prime)$, one can know the probability that $p\prime$ appears in C, which leads to a prune.

Pruning prediction algorithm. The algorithm for predicting a lower bound of the number of prunes is described in Algorithm 3.

First, let us consider the probability that two points p_1 and p_2 are hashed into the same bucket. Such a probability, according to Equation 2.1, is proportional to their distance c.

Second, in our algorithm, the full distance range is cut into multiple intervals. Then, given a distance value c, function $I(c)$ can be used to determine which interval c falls in. Counter $Count[]$ is a histogram for storing the number of points in a particular interval. The interval is determined by the collision probabilities of pairwise distances. For a fixed parameter r in Equation 2.1 the probability of collision $P(c)$ decreases monotonically with $c = ||p_1 p_2||_p$, where p is 2 in our case where we consider *Gaussian* distribution. The optimal value for r depends on the data set and the query point, but it was suggested in [DIIM04] that $r = 4$ provides good results, and, therefore, we currently use the value $r = 4$ in our implementation.

Under this hash function setting, there is not much difference for distances c within the range $[0, 1]$ in terms of hash collision probability. Thus, we use fewer intervals. In

Algorithm 3: Predict the number of prunes

```
Input: A set $C$ with $|C|$ $d$-dimensional points; a distance threshold $\tau$ defining
       near neighbors; a query point $q$; a distance interval function $I(c)$; the
       set of similar point pairs $SP$
Output: number of prunes
begin
    Construct a sample set $S$ by selecting $s$ points from $C$ uniformly at random;
    for each point $p$ in $S$ do
        Compute the distance $d(q, p)$;
        search for $p\prime$ in $SP$ where $d(p\prime, p) < |d(q, p) - \tau|$;
        if found then
            increment the counter $Count[I(d(q, p) + |d(q, p) - \tau|)]$ by 1;
    Scale up the non-zero elements in the counter by a factor of $|C|/|S|$ and
    store them back to Count[]; PruneNumber = 0;
    for each non-zero element in Count[i] do
        PruneNumber += $Count[i]$ * $P(c_i)$;
        /* $c_i$ is the maximum distance of interval i, $P(c_i)$ is the probability that
        2 points with distance $c_i$ are hashed to the same value by the LSH hash
        function, according to Equation 2.1 */
    Output PruneNumber;
end
```

contrast, for distances within the range $[1, 2.5]$, the hash collision probabilities differ significantly. We use more intervals for this distance range.

The policy we use to cut distance range into intervals is as follows. Fix the number of intervals first (e.g., 100); assign one interval for range $[0, x_1]$ and one for range $[x_2, \infty]$, within both of which the hash collision probabilities are similar; assign the rest of intervals to range $[x_1, x_2]$ by cutting the range evenly. Since the function $I(c)$ is only determined by the hash function, the intervals cutting can be done off-line before processing the data set.

Sampling accuracy. SimPair LSH only takes a small fraction (e.g., 10%) of points from C; scaling up the number of points in the sample whose distances to q are within an interval and estimating the value in C will generate some errors.

Lemma 1 *A uniform random sample gives an unbiased estimate for the number of points with certain property, and the relative accuracy is inversely proportional to the number of points, and proportional to the sample size and the true number being estimated. The standard deviation of the ratio between the estimate and the true value is* $\sqrt{\frac{n}{R}(\frac{1}{x} - \frac{1}{n})}$*, where* n *is the total number of points to be sampled,* R *is the sample size and* x *is the true value to be estimated, i.e. the number of points with the property.*

Proof 2 *Assume R different points are randomly taken into the sample, and each point with the property has a probability $\frac{x}{n}$ to stay in the sample. Let X_i be an indicator random variable indicating if the ith point being sampled is with the property or not. That is,*

$$X_i = \begin{cases} 1, & \textit{the sampled point has the property,} \\ 0, & \textit{otherwise.} \end{cases}$$

It is not hard to see that:

$$Pr(X_i = 1) = \frac{x}{n};$$

and

$$Pr(X_i = 0) = 1 - \frac{x}{n}$$

Let

$$Y = \frac{n}{R} \sum_{i=1}^{R} X_i$$

then Y is the observed value. Since

$$E[Y] = \frac{n}{R} \sum_{i=1}^{R} E[X_i] = x$$

Y is an unbiased estimate. Thus:

$$VAR[Y] = \frac{n^2}{R^2} VAR\left[\sum_{i=1}^{R} X_i\right] = \frac{n}{R} x \left(1 - \frac{x}{n}\right)$$

2.4 Experiments and Evaluations

In this section, we demonstrate the practical performance of our approach on three real-world data sets, testing the pruning effectiveness, pruning costs, real running time and memory saving, pruning prediction and quality of results from SimPair LSH.

2.4.1 Performance metric

The following paragraphs describe the metric we use to evaluate the performance of our approach: number of prunes (distance computations saved), number of operations to achieve the saving, real running time and memory consumption.

Number of prunes. To measure the effectiveness of our approach, we use the number of pruned points (number of prunes in short) as one of the main metrics. The number of prunes is a metrics independent of the specific implementation.

Number of operations to achieve the pruning. To measure the cost for obtaining the pruning, we monitor the number of operations including all CPU operations (e.g., addition, multiplication, comparison) and memory accesses in searching for p in SP and looking up $p\prime$ from C. These operations may not cost exactly the same amount of time, but they can be considered as a performance indicator independent of the specific implementation.

Real query response time saved. We also give the results of real running time to show the performance gain in clock time. Because the time spent on generating hash values and retrieving the candidate set C is always the same for both SimPair LSH and the original LSH, and the time percentage of this portion within the overall running time varies significantly with the parameter setting of L and the size of C (the latter depends on the queries), we only consider the time spent on finding near neighbors from C, which is $O(d * |C|)$ for LSH.

Note that since our algorithm with the pruning process reduces the number of points in C, the gain in running time cannot exceed 100% because it is related to the percentage of the pruned points. For example, having 30% of prunes, we expect a gain in time at most of 30%, but in reality it will be less, due to the costs SimPair LSH has to pay for gaining the pruning.

Memory consumption. Since memory size is the main constraint for LSH-based similarity search, we compare the memory costs of different approaches to reach the same response time. In addition to the space used for storing LSH indices as in the original LSH approach, we use a small amount of extra space for SP storing the similar point pairs taking at most 10% of the index size.

2.4.2 Data sets

We use three real-world image data sets in our experiments: one directly downloaded from a public website and two generated by crawling commercial multimedia websites.

Flickr images. We sent 26 random queries to Flickr [Fli] and retrieved all the images within the result set. After removing all the images with less than 150 pixels, we obtained approximately $55,000$ images.

Tiny images. We downloaded a publicly available data set with 1 million tiny images [TFF07]. The images were collected from online search tools by sending words as queries, and the first 30 returned images for each query are stored. Due to the high memory cost of LSH for large data sets, we picked $50,000$ images uniformly at random from this 1 million tiny image data set. This random selection operation also reduced the chance that similar pairs appear in the data set since the images retrieved from the result set of one query have higher chance to be similar to each other.

The reason why we used this smaller data set rather than only considering the full set was that we could vary the number of hash tables within a larger range and observe the behavior of the algorithms under different number of hash tables.

For example, the largest number of hash tables we used was about $1,000$; indexing 1 million data points takes 12GB memory under this setting which was above the memory limit of our machine. (Note that this is an extreme case for experimental purpose and may not be necessary in practice.) If we used 10 hash tables, then the memory consumption will drop to 120MB. We also conducted experiments on the whole 1 million data set setting the number of hash tables to smaller values, so as to see how the algorithms behave with a larger data set size. The results from the 1 million data set will be also reported.

Video key-frames. We sent 10 random queries to Youtube [You] and obtained around 200 video clips from each result set, approximately $2,100$ short videos in total. Then, we extracted all the frames of the videos and their HSV histograms with dimensionality 162. HSV is the common cylindrical-coordinate representation of points in an RGB color model. After that, we extracted key frames of the videos in the following way: sequentially scan the HSV histogram of each frame in a video; if the euclidean distance between the current histogram and the previous one in the video is above 0.1 , keep this histogram; otherwise skip it. We set the distance threshold to 0.1 because two images with this distance are similar but one can clearly see their difference based on our observation. In the end, we obtained $165,000$ key-frame images.

For all the image data sets described above, we removed duplicates and converted each image within a data set into a d-dimensional vector ($d = 162, 512$) by using the standard HSV histogram methods [GW07]. Each entry of the vector represents the percentage of pixels in a given HSV interval.

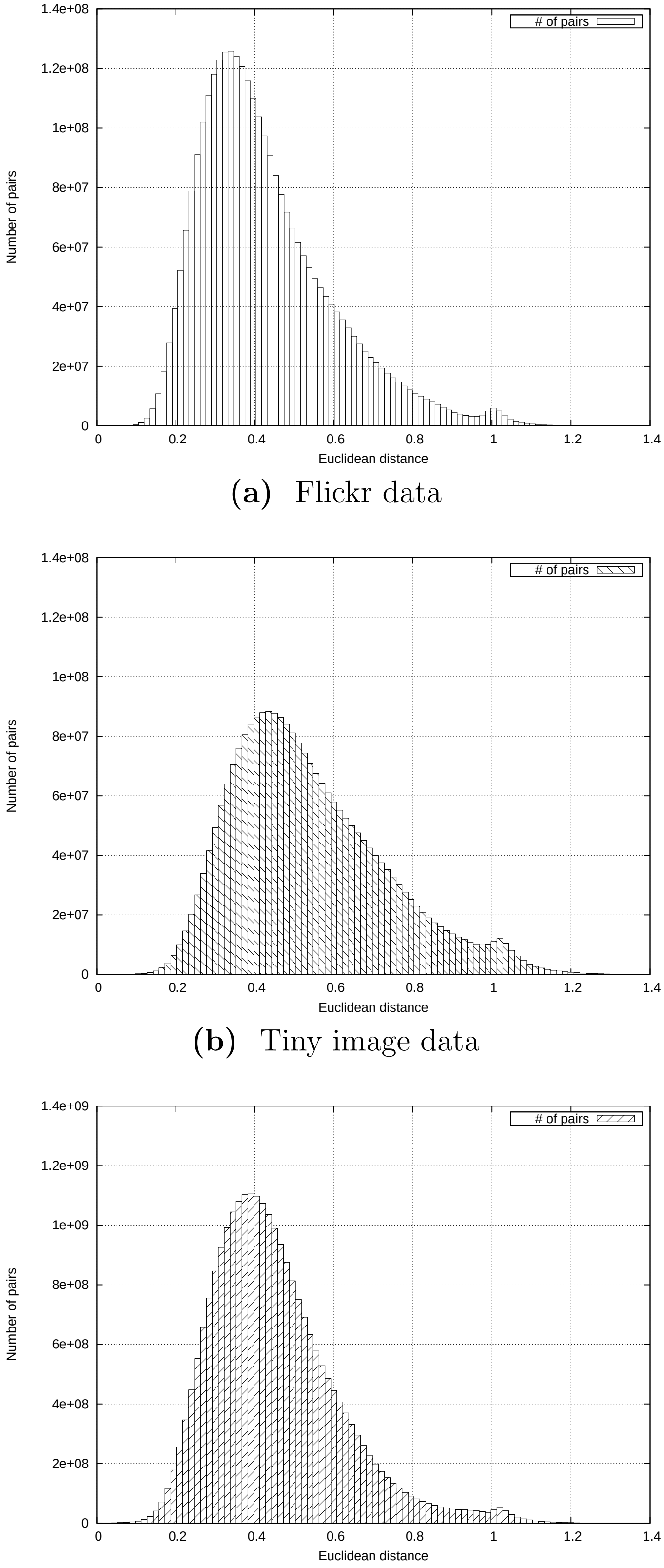
Pair-wise distance distribution. Since the pair-wise distance distribution of data set may affect the result of our experiments, we show them here to understand the data set better.

We cut the distance range into multiple intervals and count the number of points within each interval. The histograms are shown in Figures 2.1a, 2.1b, 2.1c respectively for the Flickr image, Tiny image and Video key-frame data sets. It might be interesting to see that the distributions of these data sets look similar.

2.4.3 Experimental setup

All experiments were ran on a machine with an Intel T2500 2GHz processor, 2GB memory under OS Fedora 9. The algorithms were all implemented in C compiled by gcc 4.3.0.

The data points and the LSH indices are both loaded into the main memory. Each index entry for a point takes 12 bytes memory. To test the performance of our approach, we randomly selected a certain number of objects from the data set as query objects, and measure metrics as discussed before. We took the average number of pruned points of all queries, the average percentage of pruned points, and



(a) Flickr data

(b) Tiny image data

(c) Video key-frame data

Figure 2.1 Number of point pairs vs. distance intervals

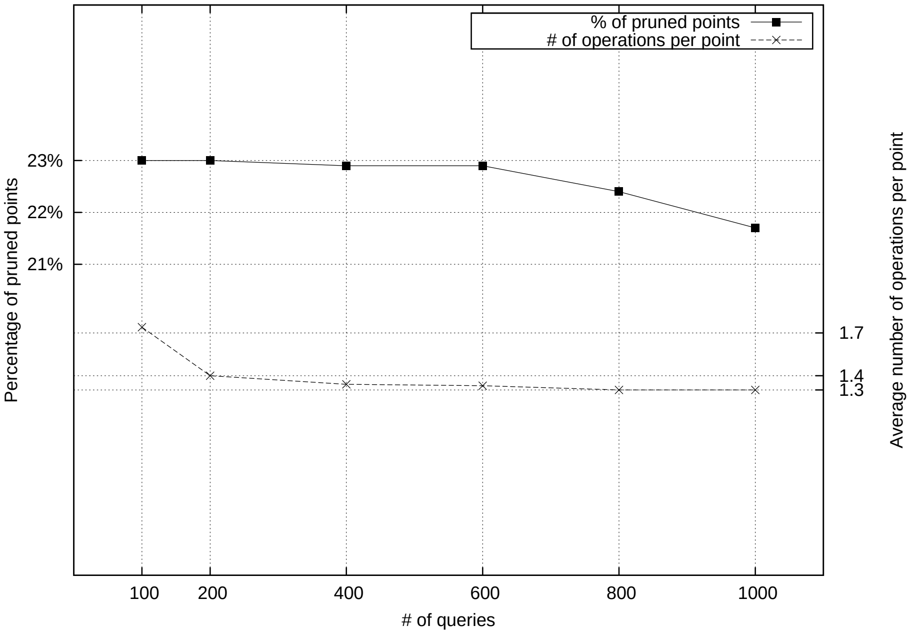


Figure 2.2 Percentage of prunes and number of operations to gain the pruning vs. number of queries

the average number of operations spent on achieving the pruning per point (average number of cost operations per query/average size of candidate sets C of all queries).

2.4.4 Experiments testing pruning effectiveness and costs

In this set of experiments, we tested the number of distance computations saved by our approach, the time and space cost to obtain the saving. We used the Flickr data set, and the results from other data sets are also consistent in general.

Varying the number of queries. First, we randomly selected a certain number of queries, and varied this number from 100 to 1000 to see if the pruning gain and costs are sensitive to the number of queries. Note that we only consider the queries which generate a candidate set C with more than T points, as described in the next experiment, because there is not much necessity to start the pruning process if $|C|$ is small. The parameter settings are as follows: distance threshold for near neighbors $\tau = 0.1$, the dimensionality of the data set $d = 162$, the distance threshold maintained by the similar pair list SP $\theta = 0.1$, the number of hash tables $L = 136$, and the success probability $P = 95\%$. The parameters will be the same in the following experiments unless explicitly specified.

The results are shown in Figure 2.2. *x-axis* indicates the number of queries; the left *y-axis* shows the average number of pruned points in C of all queries; the right *y-axis* shows the average number of operations per point spent on achieving the pruning.

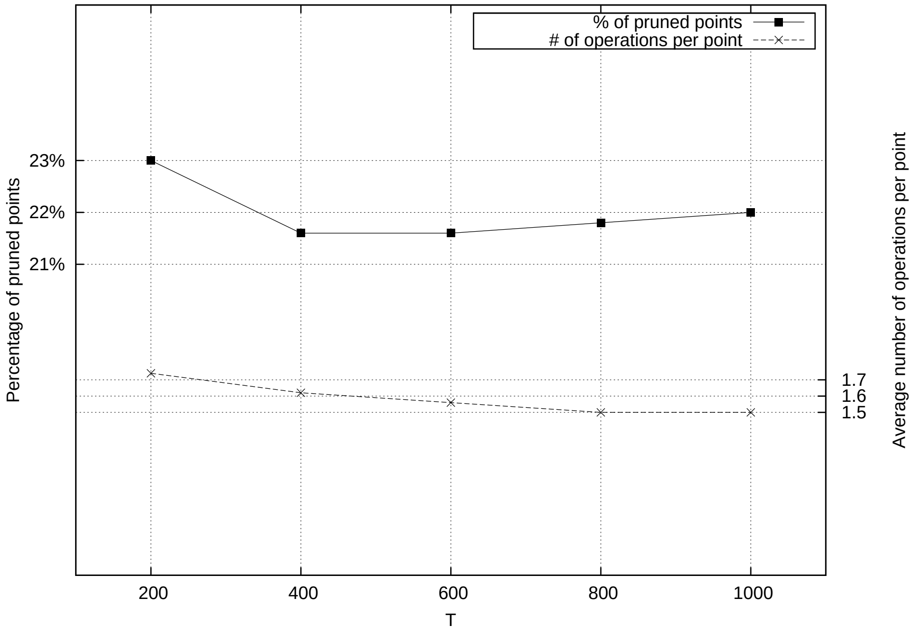


Figure 2.3 Number of prunes and costs vs. candidate set size cut-off threshold T

From the figure we can see that the algorithm performance is not sensitive to the number of queries, therefore we fixed the number of queries to 100 in the rest of the experiments.

Varying the candidate set size cut–off threshold T. Since candidate set size $|C|$ for some queries is quite small (e.g., < 50), and there is no need to start the pruning process, we set a cut–off threshold T for $|C|$. When $|C| > T$, SimPair LSH starts the pruning process; otherwise, SimPair LSH does not start the pruning process and degrades to the original LSH. We varied the threshold T to see if it has significant impact on the pruning and cost (i.e. average number of operations per point). The number of queries is set to 100 and other parameters are the same as the previous experiment. The results are shown in Figure 2.3. The left *y-axis* shows the percentage of pruned points; the right *y-axis* shows the number of cost operations per point to gain the pruning.

From the figure we can see that the algorithm performance is not sensitive to T in terms of both the percentage of pruned points and the average number of operations cost per point. In the rest of the experiments, we fixed T to 200.

Varying the distance threshold of near neighbors τ. We varied the value of τ to see how τ affects the pruning and costs. Other parameters are the same as the previous experiment. The results are shown in Figure 2.4. *Y-axises* are similar to the previous figures while *x-axis* is τ.

From the figure we can see that τ has no significant impact on the pruning effec-

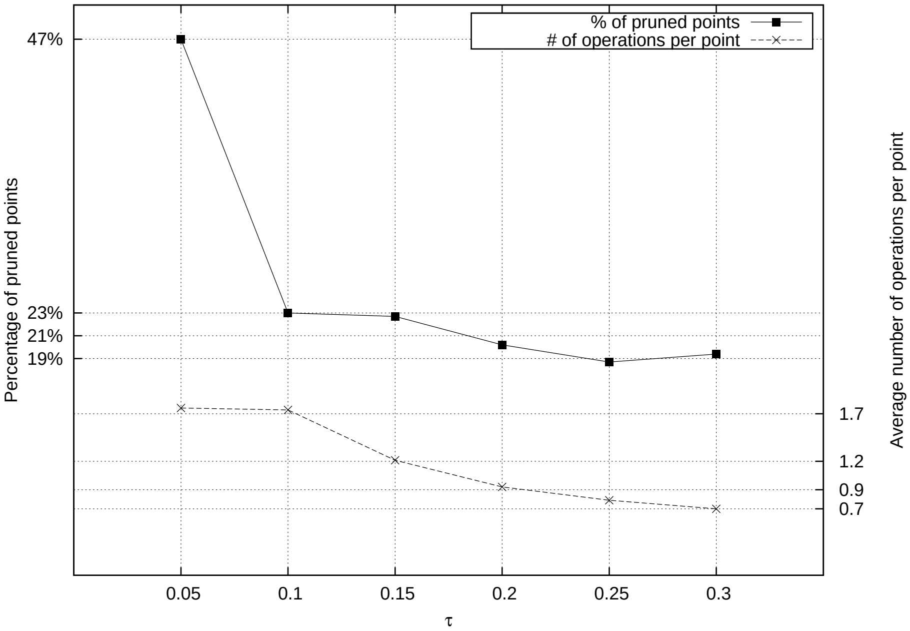


Figure 2.4 Number of prunes and costs vs. τ

tiveness either. The high percentage of pruned points when $\tau = 0.05$ is because the size of the candidate set is small, and the variance of the percentage is higher. As previously presented, we fixed $\tau = 0.1$ in the rest of the experiments.

Varying the dimensionality d**.** We tried two different dimensionalities d: 162 and 512 to see how d affects the pruning and costs. Other parameters are the same as the previous experiments. The results are shown in Figure 2.5. The left *y-axis* shows the percentage of pruned points as in the previous experiments. The numbers on top of the bars in the figure show the average cost operations per point. From the figure we can see that the cost operations is larger when the dimensionality is higher. This is partially because the number of similar pairs in SP is relatively larger when d is large, which has a similar effect as increasing θ (when d is large, there are more similar pairs although the similarity threshold does not change).

To see the real query time saved, we have another figure in the next sub-section.

2.4.5 Experiments testing the query response time

In this set of experiments, we report the query response time of the original LSH indexing and our approach. The LSH code were obtained from Andoni [AI05], and we conducted the experiments for LSH without changing the original source code.

Hash function time costs. Note that during query time, generating the hash values of each query also takes time where the amount depends on the number of hash functions or hash tables used. In the case that the candidate set size is relatively small

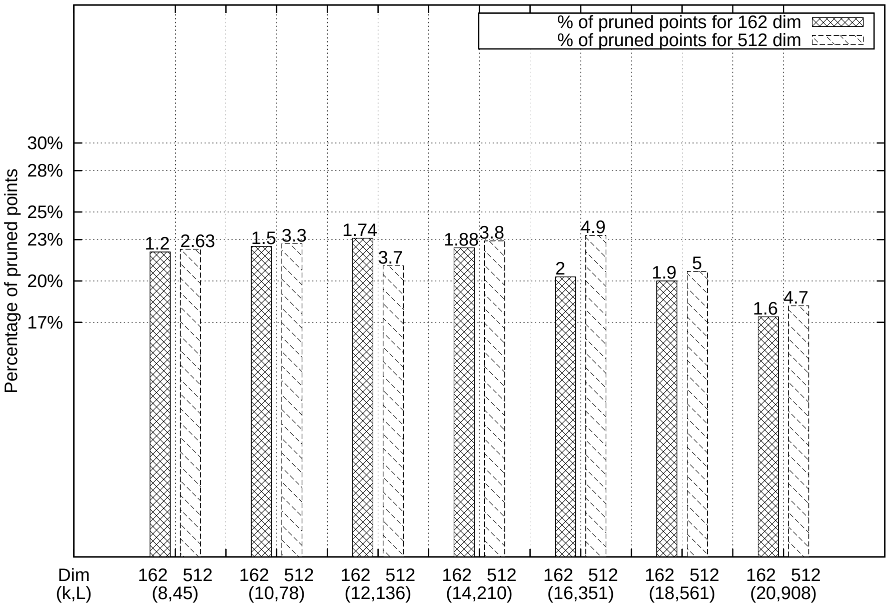


Figure 2.5 Number of prunes and costs vs. data dimensionality d

and the number of hash functions is large, the hashing process can take as long as the time spent on scanning the candidate set. Since the time spent on generating the hash values is exactly the same for both SimPair LSH and the original LSH, and the percentage of this portion with respect to the overall time varies significantly with the parameter setting of L and the size of C which depends on the queries, we only report the time spent on finding near neighbors from C to see the difference between our approach and LSH better. When the size of C is relatively large, the fraction of hash function time cost is relatively small. But in the worst case where hashing queries take the same amount of time as scanning C, the time difference between two approaches will be half of the numbers reported hereafter .

Varying k and L. We varied the hash function parameter k and the number of hash tables L to see how these parameters affect the query time. τ was set to 0.1, success probability P was set to 95%. Note that once k was fixed, the number of hash tables L was also fixed to guarantee the required success probability. Other parameters are the same as in the previous experiments. The results on the 3 data sets are shown in Figure 2.6. *Y-axis* is the response time saved by SimPair LSH computed as follows: (LSH Time - SimPair LSH Time)/(LSH time).

From the figure we can see that SimPair LSH consistently outperforms LSH under different settings of k and L. The extra memory consumptions of the full similar pair set SP when $\theta = 0.1$ are 73.7MB, 12.5MB, and 3.7MB respectively for the video key frame, Tiny image and Flickr image data sets. Recall that SP size is bound to the 10% of the index size, thus when L is small, we use a smaller θ. Since LSH can also

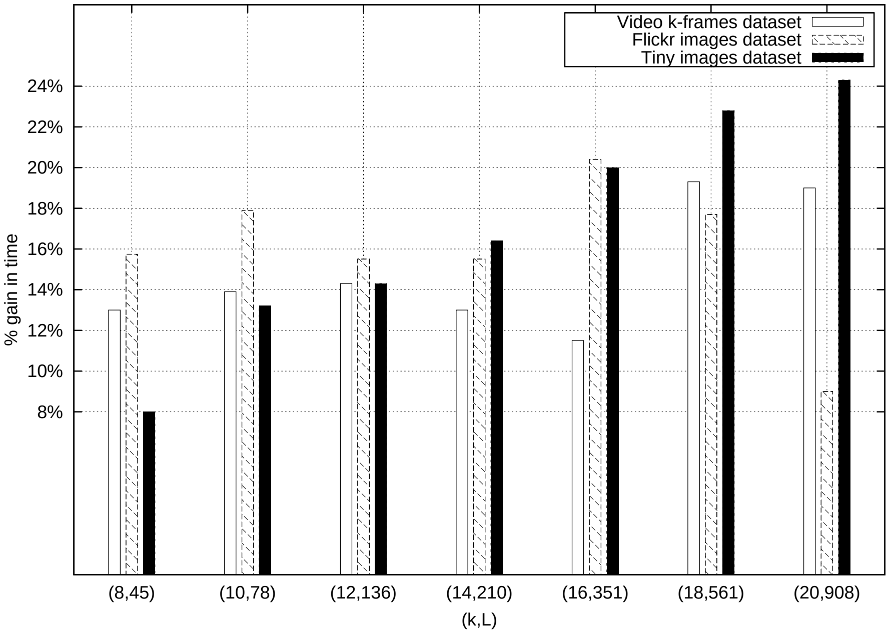


Figure 2.6 Running time vs. LSH parameters (k, L)

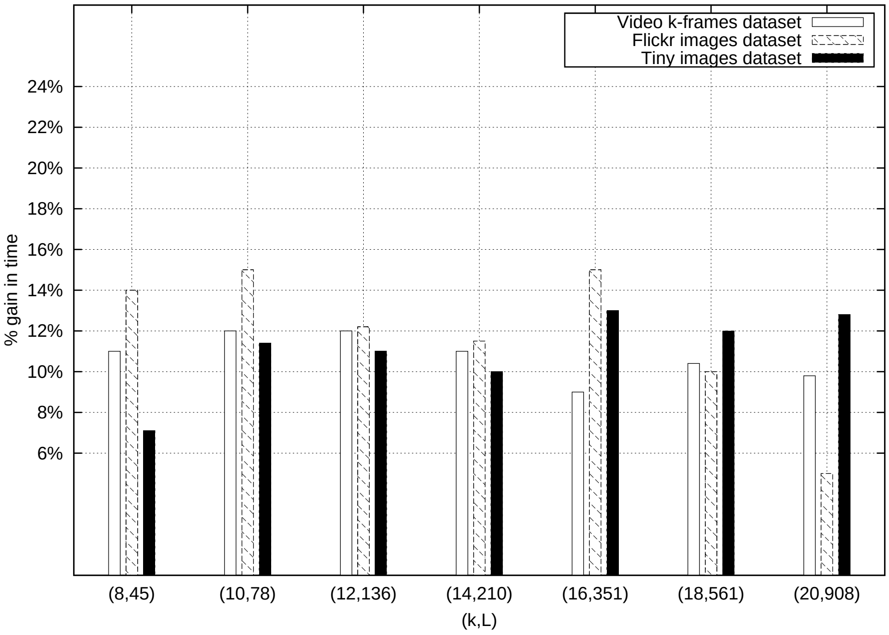


Figure 2.7 Overall running time vs. LSH parameters (k, L)

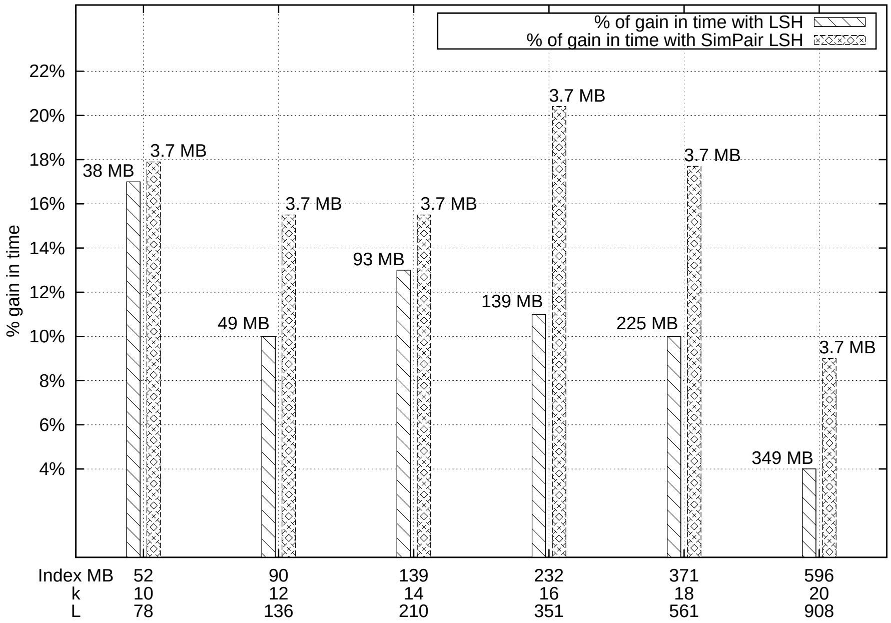


Figure 2.8 Running time saved vs. LSH memory consumption

save running time by increasing the number of hash tables (varying k and L), we tested how much additional memory LSH needs to gain the same amount of time in the next experiments.

For completeness in Figure 2.7 we report the response time saved by SimPair LSH considering the overall running time.

Extra space cost comparison showing the significance of our time gain. To achieve the response time gain, in addition to the memory cost of the LSH indices, the extra cost SimPair LSH takes is the memory spent on the similar pair list that we restricted at most to a constant fraction of the LSH indices (10% in our experiments). To achieve approximately the same running time gain, LSH can also increase the memory consumption by increasing k and L without resorting to our approach. Therefore, we compared the memory consumption of the two approaches to achieve roughly the same query time improvement. In fact, the memory cost of LSH can be computed from L: each hash table stores the identifiers of all n points, and each identifier takes 12 bytes as implemented by Andoni and Indyk [AI05]; therefore, the LSH space cost is $12 * n * L$ while the LSH's time gain is computed from $12 * n * (L_{i+1} L_i)$ where i represents the i-th setting.

Hence, to see the size of extra space LSH needs to achieve roughly the same SimPair LSH's query time improvement, it suffices to check the value of L. Since the size of C dominates the time LSH scans through the candidate set, the running time being saved can be represented by the reduction of $|C|$. Recall that bigger values of L correspond to the decrease in size of C.

The Figure 2.8 is based on the Flickr image data set. The *x-axis* presents the memory consumption of hash tables needed to index all the points in the dataset for diverse settings of k and L; for Flickr image data set $n = 55,000$. Since SimPair LSH uses the indices of LSH, this memory utilization is common for both algorithms. The *y-axis* shows the percentage gain in time. The numbers on top of the bars show the extra memory cost required to reach the gain in time reported. It is important to note that the extra memory cost for SimPair LSH remains constant to $3.7MB$ (i.e. the similar pair list size) for all the diverse setting of L, while LSH needs significantly more extra memory, with increasing L, to achieve similar response time as our method. We can conclude that LSH needs significantly more memory to achieve even less response time saving as SimPair LSH does.

For example, to have a gain in running time of 17% when $L = 78$, LSH needs $38MB$ extra memory $(12 * 55,000 * (136 - 78))$. In contrast, SimPair LSH only needs $3.7MB$ storing the similar pair set to gain more than 17% response time; that makes our algorithm save 10 times the space cost used from LSH. Note that when the number of hash tables is increased, the time spent on computing the L hash values will also increase proportionally, which means the real running time saving from Original LSH is actually smaller than 17%. When L is large, even more extra memory is needed to gain the same amount of running time. For example, when the hash table size increased by around $225MB$, the time cost decreases only about 10%; in this case our method saves roughly 60 times the space cost used from LSH.

Comparing with the figure shown in the previous experiment, by using the same amount of extra memory ($3.7MB$), SimPair LSH gains slightly more percentage for different settings of k and L. Clearly, SimPair LSH is more space efficient in terms of saving running time.

Varying the success probability P. We varied P from 90% to 99%, and set $k = 14$ and L changes accordingly to see how P affects the real running time. Other parameters are the same as the previous experiment. The results are shown in Figure 2.9. Again from the figure, we can see that SimPair LSH outperforms LSH in terms of running time consistently. For different data sets, P has different impact on the saving time. However, the general trends seem to indicate that the impact is not significant.

For completeness in Figure 2.10 we report the response time saved by SimPair LSH considering the overall running time.

Varying the dimensionality d. We ran experiments on Flickr image data set with different dimensionality d: 162 and 512, and set $P = 95\%$ to see how d affects real running time saved. Other parameters are the same as the previous experiments. The results are shown in Figure 2.11. The *y-axis* shows the percentage of running time saved as in the previous experiments.

From this figure, we can see that for a higher dimensionality, the percentage of real time saved is higher in general, which we were not able to see from Figure 2.5.

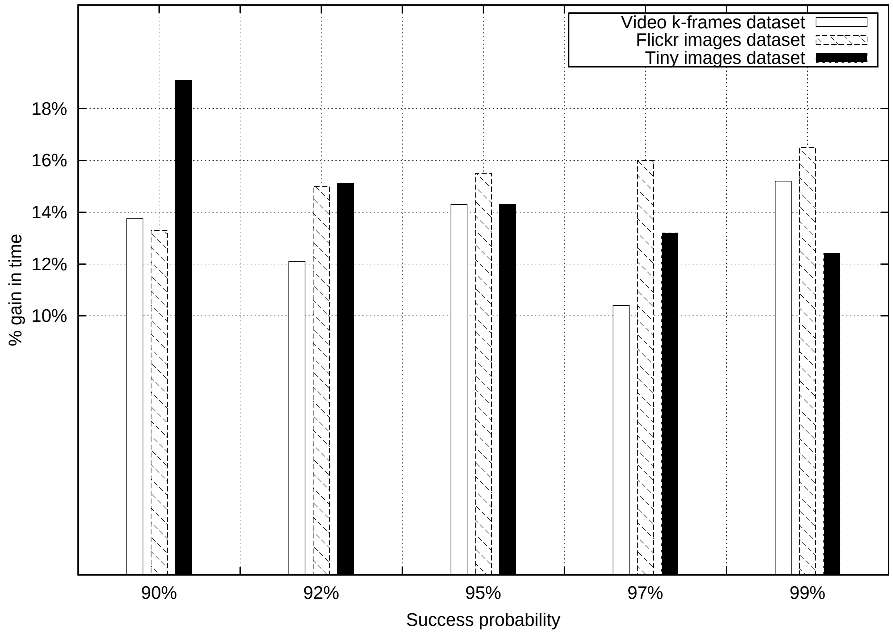


Figure 2.9 Running time vs. success probability P

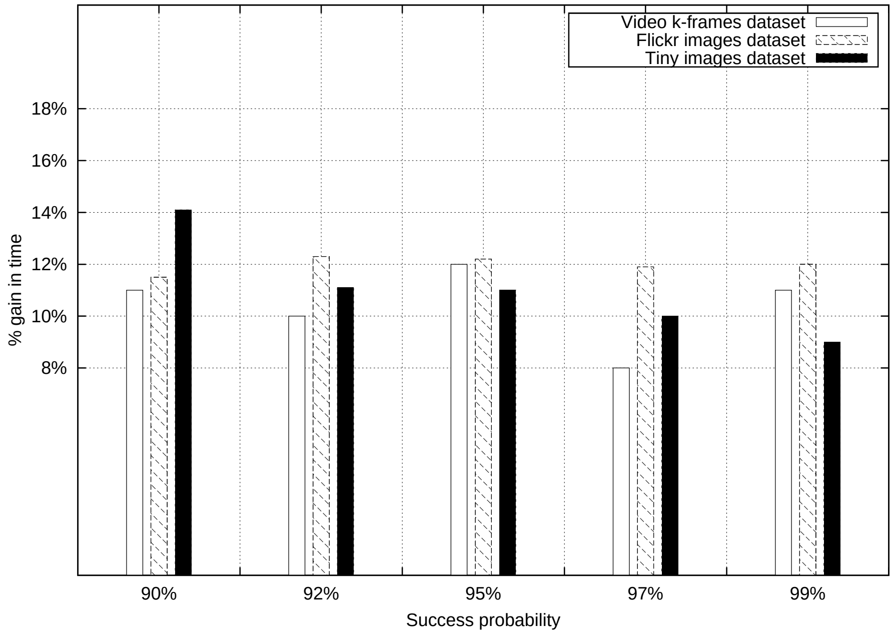


Figure 2.10 Overall running time vs. success probability P

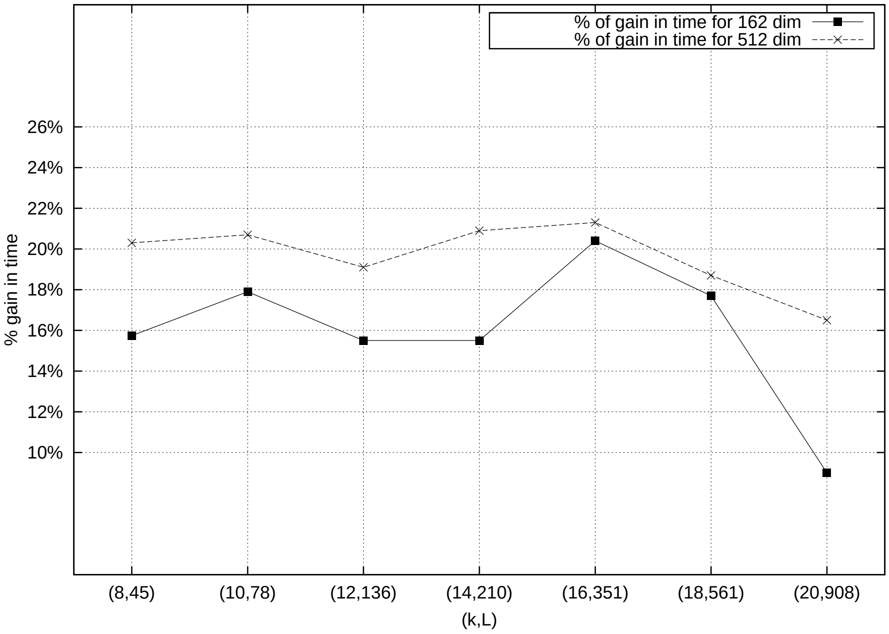


Figure 2.11 Running time vs. data dimensionality d

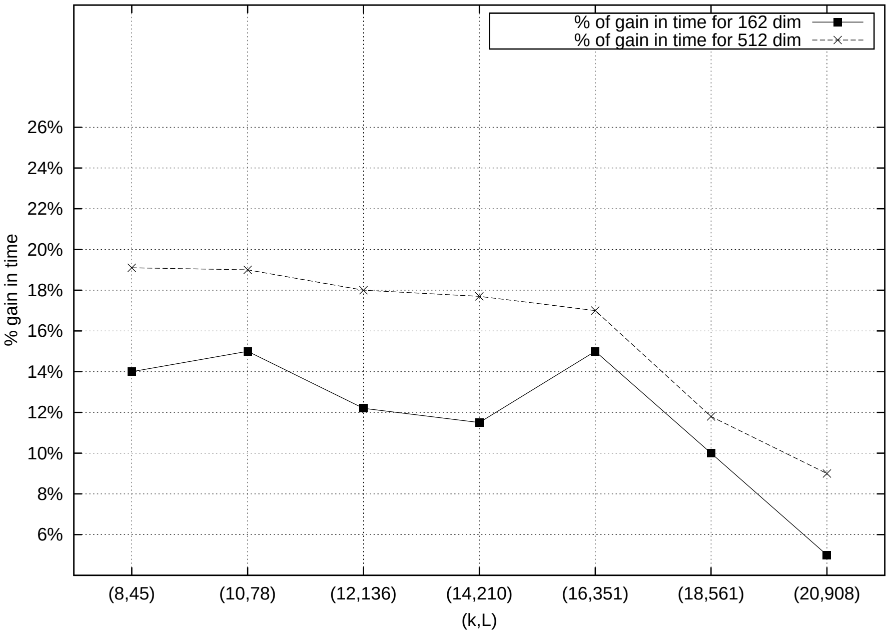


Figure 2.12 Overall real time saved vs. data dimensionality d

This is because the gain in time each prune brings is relatively higher compared with the cost of each prune when the dimensionality is higher.

For completeness in Figure 2.12 we report the response time saved by SimPair LSH considering the overall running time.

2.4.6 Experiments on larger data sets

As mentioned earlier, we also ran experiments on the full 1 million tiny image data set. Due to the memory size constraint on our machine (2GB), we set the number of hash tables to smaller values. The results were consistent with those shown before, and they showed that data set size is not a dominant factor affecting the difference between SimPair LSH and LSH. For example, referring to Table 2.2 when the number of hash tables is set to 45, SimPair LSH outperforms LSH by around 11% on the larger data set in terms of real running time, which is similar to what Fig. 2.6 showed.

Table 2.2 Running Time saved for 1 million tiny image data set

k	L	Partial Time saved	Overall Time saved
8	45	11 %	9 %
10	78	14 %	12 %

2.4.7 Experiments testing the pruning prediction

In this subsection, we test the prediction of the number of prunes and the number of operations to achieve the pruning.

We vary the similar pair threshold θ. The predicted lower bound of pruned points with different θ is shown in Figure 2.13. The parameter settings are as follows: distance threshold for near neighbors $\tau = 0.1$, the dimensionality of the data set $d = 162$, $k = 20$, $L = 595$, and the success probability $P = 90\%$. *Y-axis* shows the average number of points pruned per query; *x-axis* shows the values of θ based on τ.

From the figure we can see that the pruning prediction is relatively accurate. The difference between the predicted lower bound is in the worst case at most 50% of the number of real pruned points under different settings of θ.

2.4.8 Quality of results

In this set of experiments, we tested the recall or false negatives of both methods. If a user sets P at 90%, it means that she can tolerate missing at most 10% near

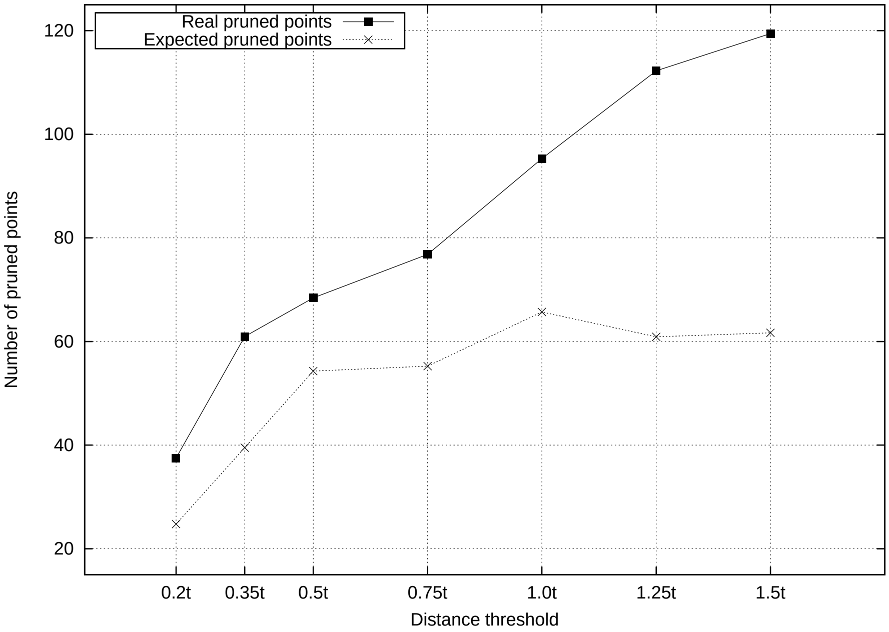


Figure 2.13 Predicted prunes vs. θ

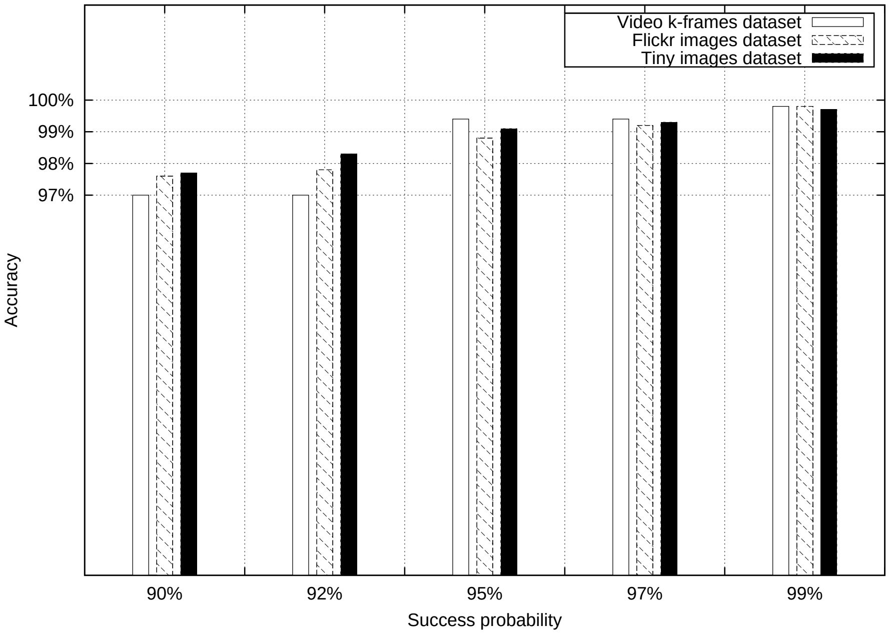


Figure 2.14 Recall (Quality of results) vs. success probability θ

neighbors. The results in Fig. 2.14 show that the real recall value is clearly higher than the user specified probability.

2.5 Conclusions

In this chapter, we studied the problem of range search in an incremental manner based on a well-known technique, Locality Sensitive Hashing. We proposed a new approach to improve the running time of LSH. The idea is to take advantage of certain number of existing similar point pairs, and checking this similar pair set on-the-fly during query time. Since the look-up time cost is much cheaper than the distance computation, especially when the dimensionality is high, our SimPair LSH approach consistently outperformed the original LSH method, with the cost of a small amount of extra space. To gain the same amount of running time, LSH required significantly more space than SimPair LSH (e.g., 10 to 100 times more). The superiority of SimPair LSH over the original LSH was confirmed by our thorough experiments conducted on 3 real-world image data sets. Furthermore, SimPair LSH preserves the theoretical guarantee on the recall of the search results. Last, SimPair LSH is easy to implement based on LSH.

3

Visualizing Technology Enhanced Learning Research Communities

[1]

Author Co-Citation Analysis (ACA) provides a principled way of analyzing research communities, based on how often authors are cited together in scientific publications. In this work, we present preliminary results based on ACA to analyze and visualize research communities in the area of technology-enhanced learning (TEL), focusing on publicly available citation and conference information provided through CiteseerX [Cit] and DBLP [DBLa]. We describe our approach to collecting, organizing, and analyzing appropriate data, as well as the problems which have to be solved in this process. We also provide a thorough interpretation of the TEL research clusters obtained, which provide insights into these research communities. We used principal component analysis to detect appropriate clusters in TEL research, and then visualize and interpret these clusters. The results are promising, and show the method's potential as regards mapping and visualizing TEL research communities, making researchers aware of the different research communities relevant for technology enhanced learning, and thus better able to bridge communities wherever needed.

[1] Image under Creative Commons License avaliable at http://www.flickr.com/photos/unisgeneva/4887296251/sizes/l/in/photostream/

3.1 Introduction

Technology-enhanced learning is a fascinating field, with lots of different research questions and aspects to focus on. Researchers in TEL can focus on learning infrastructure to support the re-use of learning objects or personalization, on intelligent tutoring systems, on mobile learning, or on collaborative learning in teams. They can also focus on professional learning and knowledge management infrastructures, learning in universities (e.g., computer science, engineering, or other disciplines) and on learning in schools, with a lot of interesting research questions and results. Many different conferences and journals are devoted to different aspects of technology-enhanced learning, providing a variety of forums through which to publish TEL research results.

The downside of this variety is, however, that TEL is a much more fragmented area than most other research areas, making it difficult to gain an overview of recent advances in the field. Even for experienced TEL researchers answering the questions: "What communities and sub-communities can be identified in TEL", "what research topics/specialties can be identified in a field of studies" and "what conferences are the most relevant for what topic and for what community" is a difficult task, and for beginners it is obviously an impossible one.

Being aware of this fragmentation and of the various sub-communities, which make up the TEL area, is an important pre-requisite towards overcoming this fragmentation, increasing synergies between different sub-areas and researchers, and, last but not least, providing funding agencies with evidence of new research results, innovative applications, and promising new approaches for technology enhanced learning.

In this chapter, we provide a first step towards this goal, by employing the technique of Author Co–citation Analysis (ACA) on the large subset of TEL conferences related to computer science as indexed by DBLP [DBLa] and CiteseerX [Cit] - the latter provides citation information for each indexed paper. ACA relies on the insight that, if two authors are cited together very often in scientific articles, their work must be related to the same research field.

The remainder of this chapter is organized as follows: in Section 3.2 we discuss about related works. Then, we will describe our methodology for data collection, solutions for problems that we encountered, and the techniques of author-co-citation and factor analysis for detecting communities in a given research area in Section 3.3. We will further describe and discuss our results which provide an interesting insight into some important TEL research clusters in Section 3.4, and close with a summary and discussion of next steps and future work in Section 3.5.

3.1.1 Our contributions

The contributions of this work are:

- We offer a principled way of analyzing research communities based on authors cited together in scientific publications.
- We ease authors' work to find collaboration between researchers within the same scientific community, and also increasing synergies between different sub-communities and researchers. Thus, we make researchers aware of the different research communities relevant for technology enhanced learning, and then better able to bridge communities wherever needed.

3.2 Related Work

3.2.1 Principal Component Analysis (PCA)

The goal of PCA is to find a new set of dimensions that better captures the variability of the data. For instance, it is useful when researchers have obtained data on a number of variables (possibly a large number of variables), and believe that there is some redundancy in those variables. In this case, redundancy means that some of the variables are correlated with one another, possibly because they are measuring the same construct. Because of this redundancy, it could be possible to reduce the observed variables into a smaller number of principal components (artificial dimensions) that will account for most of the variance in the observed variables [Jol02].

More specifically, the first dimension is chosen to capture as much of the variability as possible. The second dimension is orthogonal to the first, and, subject to that constraint, it captures as much of the remaining variability as possible, and so on.

PCA has several appealing characteristics. First, it tends to identify the strongest patterns in the data. Hence, PCA can be used as a pattern-finding technique. Second, often most of the variability of the data can be captured by a small fraction of the total set of dimensions. Consequentially, dimensionality reduction using PCA can result in relatively low-dimensional data and might be possible to apply techniques that do not work well with high-dimensional data. Third, since the noise in the data is (hopefully) weaker than the patterns, dimensionality reduction can eliminate much of the noise. This is beneficial both for data mining and other data analysis algorithms [TSK05].

We briefly describe the mathematical basis of PCA. For more details, please refer to [Jol02].

Mathematical Details. Given an m by n data matrix D, whose m rows are data objects and whose n columns are attributes, the covariance matrix of D is the matrix S, which has entries s_{ij} defined as

$$s_{ij} = covariance(d_i, d_j)$$

In other words, s_{ij} is the covariance of the *i-th* and *j-th* attributes (columns) of the data.

The covariance of two attributes is a measure of how strongly the attributes vary together. If $i = j$, i.e. the attributes are the same, then the covariance is the attribute's variance. If the data matrix D is preprocessed so that the mean of each attribute is 0, then $S = D^T D$.

A goal of PCA is to find a transformation of data that satisfies the following properties:

- Each pair of new attributes has 0 covariance (for distinct attributes).
- The attributes are ordered with respect to how much of the variance of the data each attribute captures.
- The first attribute captures as much of the variance of the data as possible.
- Subject to the orthogonality requirement, each successive attribute captures as much of the remaining variance as possible.

A transformation of the data that has these properties can be obtained by using eigenvalue analysis of the covariance matrix. Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of S. The eigenvalues are all non-negative and can be ordered such that $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_{n-1} \geq \lambda_n$. Let $U = [u_1, \ldots, u_n]$ be the matrix of eigenvectors of S. These eigenvectors are ordered so that the *i-th* eigenvector corresponds to the *i-th* largest eigenvalue. Finally, assume that data matrix D has been preprocessed so that the mean of each attribute (column) is 0. We can make the following statements:

- The data matrix $D' = DU$ is the set of transformed data that satisfies the condition posed above.
- Each new attribute is a linear combination of the original attributes. Specifically, the weights of the linear combination for the *i-th* attribute are the components of the *i-th* eigenvector. This follows from the fact that the *j-th* column of D' is given by Du_j.
- The variance of the *i-th* new attribute is λ_i.
- The sum of the variance of the original attributes is equal to the sum of the variance of the new attributes.
- The new attributes are called *principal components*, i.e. the first new attribute is the first principal component, the second new attribute is the second principal component, and so on.

The eigenvector associated with the largest eigenvalue indicates the direction in which the data have the most variance. In other words, if all of the data vectors are projected onto the line defined by this vector, the resulting values would have the maximum variance with respect to all possible directions. The eigenvector associated with the second largest eigenvalue is the direction (orthogonal to the first eigenvector's direction) in which the data have the largest remaining variance.

In conclusion, the eigenvectors of S define a new set of axes. Indeed, PCA can be viewed as a rotation of the original coordinates axes to a new set of axes that are aligned with the variability in the data. The total variability of data is preserved, but the new attributes are now uncorrelated.

3.2.2 Co–author Analysis and Citation Analysis

Co–author analysis and citation analysis are important methods when analyzing scientific communities. In [OMD09], Ochoa et al. provide a very nice example of how such analysis can help to provide greater insight into TEL research communities and collaborations, through visualizing and intuitively describing research community structure, focusing on TEL publications presented at recent ED-Media conferences. They focus on co-author analysis and visualization of these relations and provide interesting insights into collaboration networks in the TEL area. Wild et al. [WM98] used the same data corpus for a trend analysis in the ED-Media conferences. By applying clustering techniques to the paper titles, they showed how certain technologies and approaches gained importance – including, among others, mobile learning, blended learning, portfolios, podcasts, game-based learning and assessment.

Similar introspectives have been applied to other research fields in the past. Henry et al. [HGEF] provide an analysis of the area of human computer interaction, based on the four major HCI conferences, focusing on citation analysis that use data relating to these conferences (between conferences, articles, and authors), word cloud visualizations to characterize the four conferences, and other visualizations that characterize collaboration and other networks. This research does not rely on sophisticated mathematical network analysis models but is a very good example of the power of visualization to make the structure of these networks explicit.

The approach we build upon in this work, author co-citation analysis, has not yet been used widely, despite its potential, for detecting and clustering scientific communities based on the mathematical notion of factor analysis. One of the best papers and a good introduction to this approach is the paper by White et al. [WM98]. This study presents an extensive domain analysis of a discipline – information science – in terms of 120 top-cited authors, based on their papers from 1972 to 1995, with citations retrieved from Social Scisearch via DIALOG [2]. Tables and graphics reveal the specialist nature of the discipline over 24 years, based on author co-citation analysis.

[2] http://library.dialog.com/bluesheets/

The results show an interesting split of the field into two main specialties which barely overlap, namely experimental retrieval/information retrieval and citation analysis. Included is also a dynamic analysis of the field, based on three 8–year–periods, which shows changes of authors and areas. The analysis is based on journal citations, but neglects important conferences such as the ACM SIGIR conference, the most relevant conference for the IR community. In contrast, the citation database used in our research, CiteseerX, includes all important computer science conferences and workshops, providing a broad overview of computer science as it relates to TEL.

Using similar techniques, Chaomei Chen and Les Carr [CC99] present an analysis of hypertext research based on the ACM Hypertext conference series, with papers included from 9 conferences over 10 years. About half of the citations in this series refer to papers from the same series which points to a very homogeneous research community. Again, dynamic analysis using three time periods is included. Only citations within these conference series were considered, while we include citations from all conferences. Due to their restricted focus, the factors discovered represent a finely grained view of the hypertext research area (including sub-areas such as design models, hypertext writing, open hypermedia and information visualization), while our factors represent broader research communities, centered around one or a few community-centered conferences such as Adaptive Hypermedia or AIED.

3.3 Collecting Co-Citation Data

Following White et al. [WM98], we assume that citing practices in a research community reflect the judgments as to which works by which authors are the most influential – for the field in general and for specific sub-themes. Aggregated over time, a definite structure emerges that can be considered the current state of the field. *Co–citation* is a very good way of establishing relations between authors that correspond to specific sub-themes and research areas in a research community – even though they do not directly reference each other. We consider author A and B to be co–cited, if they are both cited by an author C – that is, both names appear at least once in the reference section of C's paper. The more co-citations, the stronger the relationship is.

Our data sets were obtained from CiteSeerX and DBLP. CiteSeerX is a digital library focusing on the literature in computer and information science, being fairly complete. The articles are crawled automatically from the Web and then metadata and citations are extracted from these articles, again automatically. The CiteSeerX dataset contains more than 1.4 million paper records correlated with about 28 million citations. Due to the automatic data collection process, metadata in CiteSeerX are not always prefect, which leads to considerable problems that have to be solved before the analysis starts. We will describe these problems and our solutions in the following subsections. In addition, DBLP is a computer science bibliography database, which relies more on human input (the maintainer of DBLP is Michael Ley, from the

University of Trier), which covers about the same field as CiteSeerX, and currently contains about 1.3 million bibliographical records. DBLP metadata do not include citations, but has been used in our project to contribute high-quality metadata, to cope with ambiguous author names and to provide reliable conference statistics.

3.3.1 Data collection

While it was not the goal of our research to determine the most relevant authors in TEL – such a goal would involve a more elaborate discussion on how "most relevant authors" should be defined – a good sample of highly cited authors in TEL covering as many areas of TEL as possible was obviously necessary. Obtaining such a sample for a diverse area such as TEL is no trivial matter. The following paragraphs discuss our approach and the steps needed to gather such a sample. Our data collection focused on data available through the CiteseerX and the DBLP databases, both covering all computer science related researches, and will extend this through additional databases covering educational and psychological research for TEL in the future.

Obtaining a first sample. To obtain a first sample of TEL conferences, we collected the lists of TEL conferences and journals to which a small sample of 13 well–known researchers submitted their papers (i.e. Duval, Scott, Brusilovsky, Koper, Kieslinger, Klamma, Nejdl, Balacheff, Sharples, Davis, Zimmermann, Wolpers, Sutherland). From these conferences and journals (as identified in DBLP), we extracted the 100 most prolific researchers. In a second iteration, we collected the list of top–100 conferences and journals to which these 100 most prolific authors submit their papers. Our final sample of authors represents the most prolific authors from the 20 conferences and journals in the latter list that have a specific focus on TEL [3]. These conferences and journals cover 13, 557 publications in total.

For these authors we created a co–citation matrix. This first step resulted in a rather sparse matrix (with some authors not co-cited with any other authors) and consequently a set of clusters extracted through our SPSS factor analysis which was difficult to interpret. Thus, subsequent iterations were designed to extend and refine the set of authors, as discussed in what follows; in addition they included other conferences such as Adaptive Hypermedia, User Modeling or Artificial Intelligence, which provide techniques for TEL infrastructures and algorithms.

Adding more authors, increasing co-citations. As regards extending and refining the set of authors, in the second iteration we first included more authors: the 50 most prolific authors from ED-Media [EM] and EC-TEL [ECT], 15 new authors from the IEEE TLT Board and Steering Committee [BC], and 5 more authors from the Telearn archive [Arc]. We also included the top-15 cited papers or books from ED-Media 2005 – 2008 [OMD09]. Second, after merging these sets, we selected the authors with at least 20 publications in CiteseerX DB and with at least 10 co-citations

[3] Other topics are computer science (27 venues), artificial intelligence (26), human-computer interaction (22) and databases (5).

in our co-citation matrix. We also experimented with a threshold of 20 and 30 co–citations, but finally kept the 10 co–citation threshold, as the clusters obtained were of similar quality.

Disambiguating authors. At this point, we realized there was a problem of disambiguation for some names, so we decided to check the name occurrences in DBLP (where author names are manually disambiguated by the DBLP maintainer, Michael Ley) and to keep only the author strings, that unambiguously identified the TEL authors we wanted to include). For example, we deleted John Cook because we found 269 occurrences of his surname in DBLP but, when queried by his full name we found only 12 publications in DBLP and 8 publications in CiteseerX. We deleted John Black as well, because the occurrences both in DBLP and CiteseerX were too ambiguous to correctly attribute publications or citations (e.g., John Black, John A. Black, John B. Black, John D. Black, John E. Black, John R. Black, John A. Black Jr). Based on this disambiguation, we kept the full name of each author, and the initials when this did not result in duplicates or ambiguity in DBLP. This left us with 77 authors for our analysis.

Adding and checking more conferences. To better characterize the clusters found through component analysis, we checked the top 4 venues for each author. This had to be done using DBLP, as CiteseerX does not contain complete references for all papers, but sometimes only refers to them as technical reports. We then used DBLPVis [DBLb], to check for the five most prolific authors in all these TEL conferences covered by DBLP and CiteseerX (AIED, CSCW , EC-TEL, Edutainment, ICALT, ICCE, ICWL, ITiCSE, ITS (Intelligent Tutoring Systems), SIGCSE, Wissensmanagement, WMTE), to make our final co-citation matrix more complete, in total 55 authors. Using a threshold of 50 DBLP publications, we kept 30 of them. 25 of them were already in our matrix, which was an encouraging sign that our previous iterations had already produced a good sample for these TEL conferences. We added 5 new authors to our matrix, for a final matrix of 82 highly cited and co–cited TEL authors.

3.3.2 Data processing – Problems and Solutions

We conducted our analysis on CiteseerX dataset. The following paragraphs discuss our approach and give an overview about the relevant tables considered from the database, as well as the problems encountered during data processing and our solutions for these problems.

Tables. CiteseerX is organized in terms of three main tables: Papers, Authors, and Citations. The *Papers* table contains all the papers, unequivocally retrieved through an identifier. Every paper can be a different version of the same publication, each associated to a single value of the attribute *cluster*, e.g., one cluster ID is coupled with several paper IDs. In addition, the papers are connected with their authors. A single author can have multiple occurrences in the *Authors* table, one for each paper she wrote. Thus, the data set contains duplicated author identifiers, a common problem when dealing with publication data. Finally, the references for each paper are stored in the table *Citations* with the following information: paper identifier *cited–paperID* of the paper which the reference is cited by, *citation title, venue, year* and *authors* of the cited paper (a string field, with all authors concatenated).

Processing. To compute the co-citation matrix, we collected the subset of the paper citations corresponding to the references to papers written by the relevant authors, selected for our analysis. The lack of a paper identifier of the citation made our mining task more complex: to retrieve the cited papers of our author list, we had to search for our authors within the value of the attribute *authors* in the *Citations* table. This was possible after processing the dataset in three steps: 1) drop all the foreign keys inside the *Citations* table; 2) change dataset engine from InnoDB to MyISAM to enable efficient full-text search; 3) create a full-text index for the attribute *authors*. All the results were stored within a new *citations–TopAuthors* table so as to provide reasonable processing time for our queries (the size of the new table is about $50,000$ records compared to the 28 million in the original *Citations* table. Finally, to further increase processing time, we built another full-text index on authors.

Multiple author aliases. Since a single author can have multiple occurrences in the *Authors* table, we had to cope with the problem that author names may be misspelled or use initials instead of full first names; authors may also change their names or use different combinations of formal and informal names and initials in different papers, producing multiple identifiers we call *aliases* for a single person. The author “Wolfgang Nejdl” appears more than two hundreds time with his complete name, for example, and about ten times as “W. Nejdl”.

Unique author identifier. We then collected all the paper citations which had at least one previously computed alias in the *authors* attribute. For each of these, circa fifty thousand records, we added one *firstAuthor* attribute in the new table to describe a single author with aliases with one identifier, e.g., we put “Nejdl” as identifier of “Wolfgang Nejdl” and “W. Nejdl”. Thanks to the fact that *firstAuthor* contains only one identifier, we were able to solve the problem of keeping information about the identifiers of a possible second or third author who wrote the same cited paper. We therefore duplicated, for each author of interest, the corresponding citation in the new table *citations–TopAuthors* with the identifier for a second and subsequent author.

Paper multi versioning. Another issue we encountered was paper multi–versioning. Because the same paper can have several versions, each of which has been crawled from the Web and given that each of these publications keep information about their references in the *Citations* table, we had to remove from our table the duplicate citations related to different editions of the same paper. To achieve this goal, we exploited the attribute *cluster*, as described before, of the table *Papers*.

3.3.3 Matrix creation

For subsequent analysis, we then created a quadratic, symmetric matrix containing the listing of our selected authors as rows and columns, to be filled by co-citation data: for the j-th row and the i-th column, the retrieved value in this cell refers to the number of times the j-th author was co-cited with the i-th one and vice-versa.

For i equal to j we included a *null* value because it corresponds to the cell representing the number of co-citations of one author with herself.

Our matrix construction process includes three main steps:

1. Select the identifiers of all cited papers we collected in our table *citations–TopAuthors*.
2. For each of these identifiers, gather distinct authors, i.e. the values of the attribute *firstAuthor*.
3. Whenever this previously computed result set carried more than one author, for each possible author pair, we incremented the corresponding values $< i, j >$ and $< j, i >$ in the matrix.

These steps lead to the following algorithm, described in pseudo-code and relevant SQL statements in Algorithm 4.

Algorithm 4: Pseudo code for the matrix computation

```
begin
    Select distinct $cited - paperID$ from $citations - TopAuthors$;
    for each $cited - paperID$ do
        Select distinct $firstAuthor$ from $citations - TopAuthors$ where
        $cited - paperID = currentcited - paperID$;
        if more than one $firstAuthor$ then
            Compute all possible author pairs;
            for each author pair $< i, j >$ do
                Update matrix cell $< i, j >$ and $< j, i >$;
end
```

3.4 Experiments and Evaluations

We then proceeded to analyze our data, using Principal Component Analysis, to detect appropriate clusters/areas in TEL research, and then visualize and interpret these clusters.

3.4.1 Using Principal Component Analysis to Detect TEL Research Areas

Principal Component Analysis. “In the social sciences we are often trying to measure things that cannot directly be measured (so–called latent variables)”, as Andy Field states in his book [Fie09]. In our case, the interest in different topics or research areas of different authors in TEL cannot easily be measured. We could not measure motivation and interest directly, but we tried to analyze a possible underlying variable (collaboration in the form of co-citations among the major authors), to detect different sub-communities and possible trends. To do so, we used the statistical application SPSS [Fie09] to perform the Principal Component Analysis (PCA): a technique for identifying groups or clusters of variables and reduce the data set to a more manageable size while retaining as much of the original information as possible. Often, its operation can be thought of as revealing the internal structure of the data in a way which best explains the variance in the data.

PCA vs. FA. Principal Component Analysis is similar to Factor Analysis, but merely has the goal of finding linear components within the data and how a variable might contribute to these components (which basically means, finding some meaningful clusters within the data). Factor Analysis uses the same techniques, but the aim is to build a sound mathematical model from which factors are estimated. Factor Analysis assumes that the covariation in the observed variables is due to the presence of one or more latent variables (factors) that exert causal influence on observed variables. Researchers use Factor Analysis when they believe that certain latent factors exist boosting causal influence on the observed variables they are studying. Exploratory Factor Analysis helps researchers identify the number and nature of these latent factors. In contrast, Principal Component Analysis makes no assumption about an underlying causal model. PCA is simply a variable reduction procedure that (typically) results in a relatively small number of components that account for most of the variance in a set of observed variables. The choice of PCA vs. FA depends on what we hope to do with the analysis: whether we want to generalize the findings from our sample to a population, or whether we want to explore our data or test specific hypotheses [Jol02]. In our specific research, we used PCA because we wanted to explore the data with a descriptive method and apply our findings to the collected sample.

Correlation determinant. When we measure several variables with PCA, the correlation between each pair of variables can be arranged in what is known as an R-matrix: a table of correlation coefficients between variables. The existence of clusters of large correlation coefficients between subsets of variables, suggests that those variables could be measuring aspects of the same underlying dimensions. These underlying dimensions are known as factors (or latent variables). In Factor Analysis we strive to reduce this R–matrix to its underlying dimensions by looking at which variables seem to cluster together in a meaningful way. This data reduction is achieved by looking for variables that correlate highly with a group of other variables, but do not correlate with variables outside that group. Because our main aim is PCA, we did not have to worry about the correlation matrix determinant. Strictly speaking, the determinant or correlation matrix should be checked only in Factor Analysis: in pure Principal Component Analysis it is not relevant [Fie09], so that we could leave all our authors in the sample.

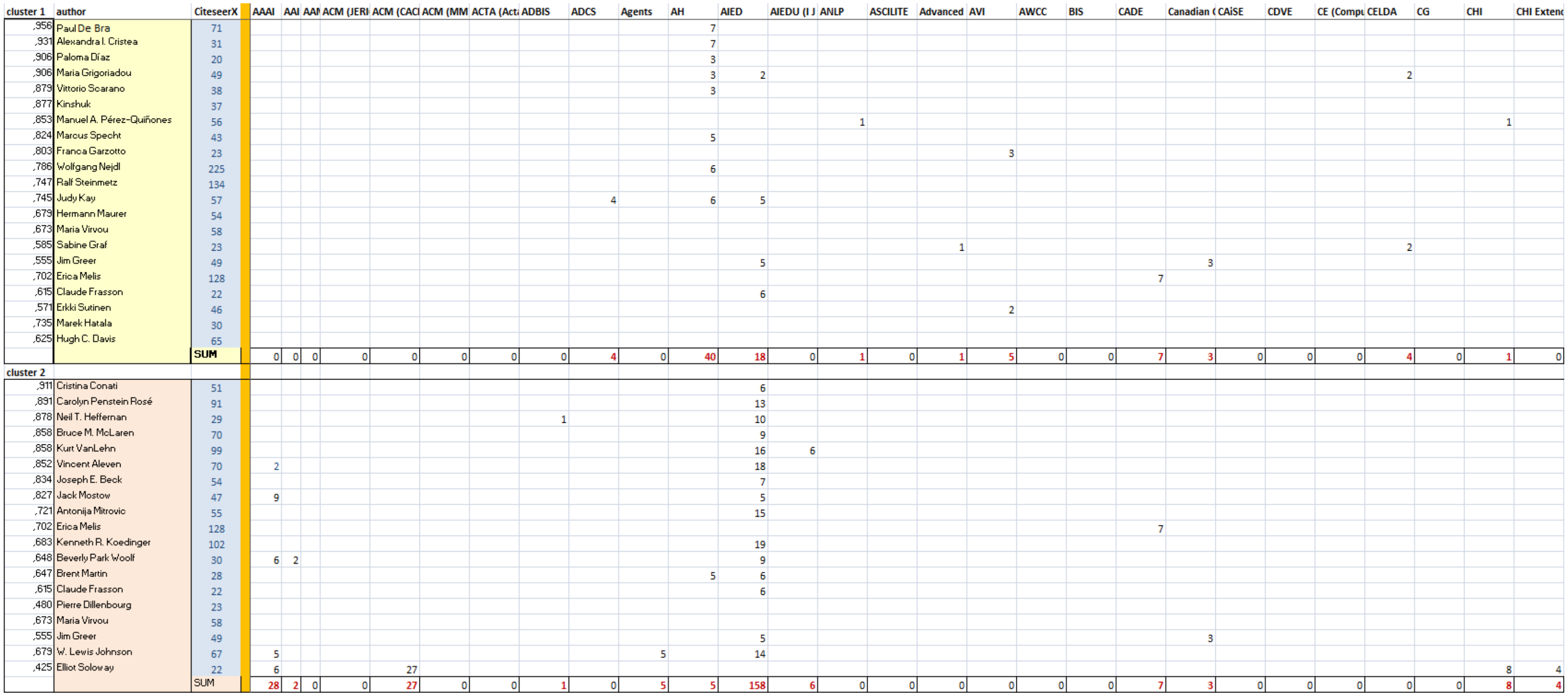
Defining factors. Not all factors are retained in an analysis, but only the most relevant and meaningful ones for the research. In our case, we used Varimax orthogonal rotation [4] to discriminate between factors (to rotate the factor axes such that variables are loaded maximally to only one factor and we could better calculate the loading of the variable on each factor). We sorted the variables by size ordering them by their factor loadings, to display all the variables which load highly onto the same factor together. As a result we obtained a Rotated Component Matrix which shows the variables listed in order of size of their factor loadings. For interpretation purposes, we also suppressed absolute values which were less than 0,4.

We obtained 15 factors in total, which explain 78% of the variance; in our work, we focus on the first six factors, explaining 59%. Compared to White and McCain's research [WM98], where the first eight factors alone explain 78% of the variance, our lower value reflects the different disciplines that come together in TEL, producing many more sub-communities, while Information Science has some well-established communities that focus on a particular topic. To describe the meaning of each factor more precisely, we also added information regarding the conferences where our sample authors usually publish. For this research, we included the top 4 venues for each author, as well as the number of papers published. Figure 3.1 shows the first two clusters, with a (small) subset of conferences displayed and Table 3.1 clusters 3–6.

3.4.2 Visualizing TEL research clusters

Visualization based on conferences. Based on this analysis, the following figures provide a visualization of the TEL research clusters obtained based on pie charts relating to the most relevant conferences for each cluster (Figure 3.2). To produce the conference-based charts, for each author we collected her four most frequented conferences according to DBLP (names of conferences as well as number of papers published by this author), added the number of papers for each conference and cluster, and then produced the following pie-charts including the most representative conferences for each cluster. For Clusters 1 and 2, conferences were selected if they included more than 20 publications (for Cluster 1) and 15 publications (for Cluster 2) from the cluster authors; for Clusters 3–6, we used a threshold of 5–7 publications to select the representative conferences.

[4] The Varimax rotation attempts to maximize the dispersion of loadings within factors. It tries to load a smaller number of variables highly onto each factor resulting in more interpretable clusters of factors.

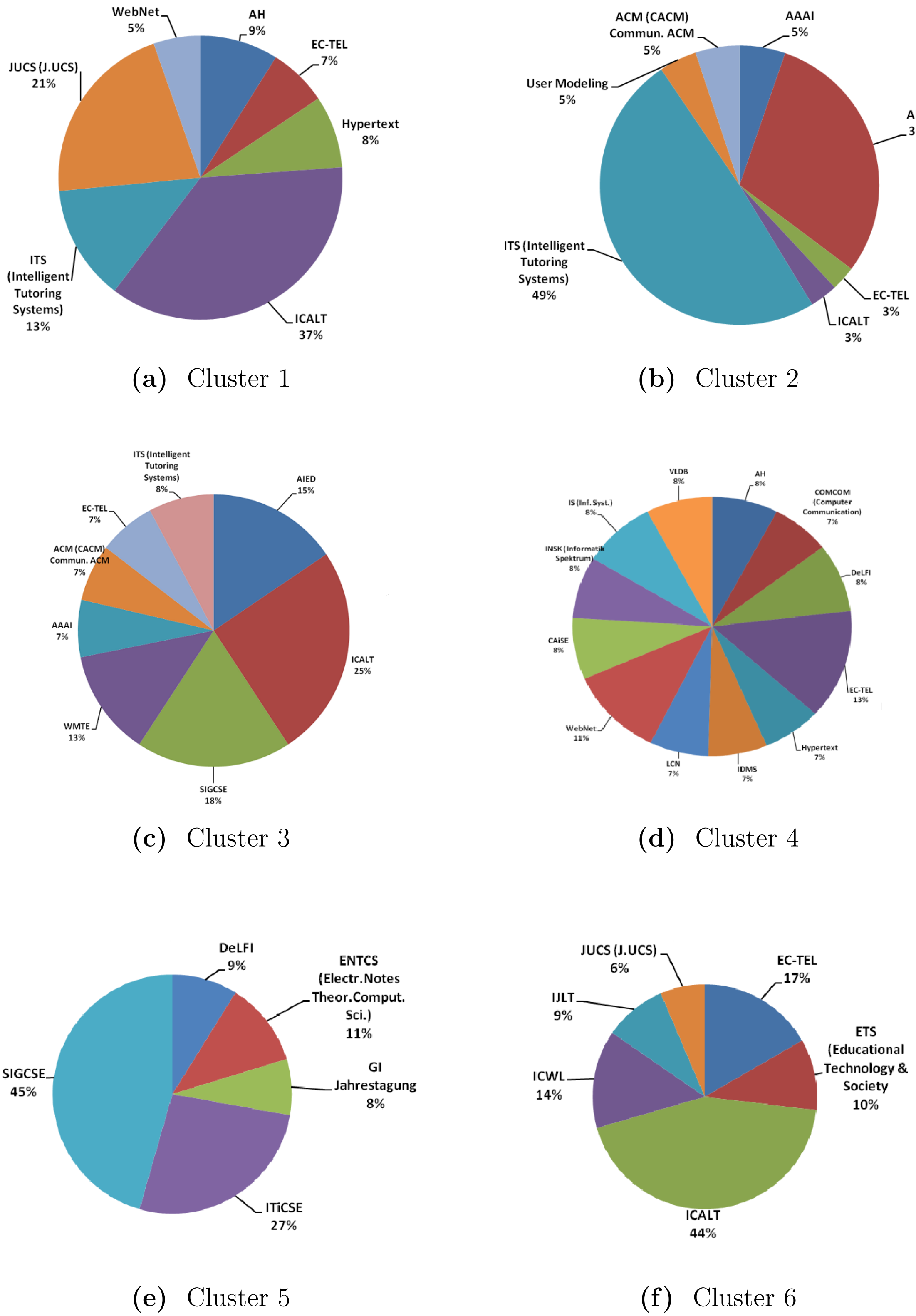


cluster 1	author	CiteseerX	AAAI	AAI	AAI	ACM (JERI	ACM (CACI	ACM (MM	ACTA (Acta	ADBIS	ADCS	Agents	AH	AIED	AIEDU (I J	ANLP	ASCILITE	Advanced	AVI	AWCC	BIS	CADE	Canadian (	CAiSE	CDVE	CE (Compu	CELDA	CG	CHI	CHI Extend
,956	Paul De Bra	71											7																	
,931	Alexandra I. Cristea	31											7																	
,906	Paloma Díaz	20											3																	
,906	Maria Grigoriadou	49											3	2													2			
,879	Vittorio Scarano	38											3																	
,877	Kinshuk	37																												
,853	Manuel A. Pérez-Quiñones	56														1													1	
,824	Marcus Specht	43											5																	
,803	Franca Garzotto	23																	3											
,786	Wolfgang Nejdl	225											6																	
,747	Ralf Steinmetz	134																												
,745	Judy Kay	57									4		6	5																
,679	Hermann Maurer	54																												
,673	Maria Virvou	58																												
,585	Sabine Graf	23																1									2			
,555	Jim Greer	49												5									3							
,702	Erica Melis	128																				7								
,615	Claude Frasson	22												6																
,571	Erkki Sutinen	46																	2											
,735	Marek Hatala	30																												
,625	Hugh C. Davis	65																												
		SUM	0	0	0	0	0	0	0	0	4	0	40	18	0	1	0	1	5	0	0	7	3	0	0	0	4	0	1	0
cluster 2																														
,911	Cristina Conati	51												6																
,891	Carolyn Penstein Rosé	91												13																
,878	Neil T. Heffernan	29								1				10																
,858	Bruce M. McLaren	70												9																
,858	Kurt VanLehn	99												16	6															
,852	Vincent Aleven	70	2											18																
,834	Joseph E. Beck	54												7																
,827	Jack Mostow	47	9											5																
,721	Antonija Mitrovic	55												15																
,702	Erica Melis	128																				7								
,683	Kenneth R. Koedinger	102												19																
,648	Beverly Park Woolf	30	6	2										9																
,647	Brent Martin	28											5	6																
,615	Claude Frasson	22												6																
,480	Pierre Dillenbourg	23																												
,673	Maria Virvou	58																												
,555	Jim Greer	49												5									3							
,679	W. Lewis Johnson	67	5									5		14																
,425	Elliot Soloway	22	6				27																						8	4
		SUM	28	2	0	0	27	0	0	1	0	5	5	158	6	0	0	0	0	0	0	7	3	0	0	0	0	0	8	4

Figure 3.1 Factor loadings, authors, overall CiteseerX publications, and top 4 venues for each author, for the first two clusters

Value	**Author**	**CiteseerX**
Cluster 3		
.923	Stefanie N. Lindstaedt	20
.919	Mark Guzdial	37
.718	Mike Sharples	38
.679	W. Lewis Johnson	67
.594	Ron Oliver	39
.571	Erkki Sutinen	46
Cluster 4		
.929	Daniel Olmedilla	45
.781	Peter Brusilovsky	93
.747	Marek Hatale	30
.735	Ralf Steinmetz	134
Cluster 5		
.911	Mordechai Ben-Ari	23
.823	Guido Roessling	32
.555	Susan H. Rodger	21
Cluster 6		
.667	Jose Luis Sierra	48
.585	Colin Tattersall	33
.583	Sabine Graf	23
.577	Rob Koper	91
.570	Baltasar Fernandez-Manjon	46

Table 3.1 Factor loadings, authors, and CiteseerX publications for cluster 3–6



(a) Cluster 1

(b) Cluster 2

(c) Cluster 3

(d) Cluster 4

(e) Cluster 5

(f) Cluster 6

Figure 3.2 A visualization of the TEL research clusters based on relevant conferences

Visualization based on Tag Clouds. Based on the clusters we retrieved, we selected within the CiteseerX dataset all the paper titles whose authors were in the cluster of interest. From the extracted paper titles we removed the words with less than 2 characters and the words consisting of numbers because these were not useful when determining the topic of a paper; for those words containing punctuation marks such as – ? % and /, we removed them and combined the remaining parts. We also removed stop words and applied stemming, as well as duplicate words inside a papers title. We then assigned a counter to each distinct word, counting the number of occurrences of the word inside the titles. Last, we sorted all words in increasing order based on the counters and visualized, in the largest case, the first 150 words (Figure 3.3).

3.4.3 Discussion

The combined information from the clusters of researchers, the main conferences and journals that they address, and the most often used keywords in their publications clearly show the differences in focus in the community – in terms of research as well as in terms of publications and connections. In this section, we discuss the main findings from the visualizations presented before.

The main publication venues (Figure 3.2) of the *first cluster* of researchers (Figure 3.1) include – besides main TEL conferences such as ITS and ICALT and the general journal JUCS – Adaptive Hypermedia, Hypertext, and EC-TEL. From the word cloud (Figure 3.3) of this cluster – with “Adapt”, “Model”, and “Hypermedia” as distinctive words – a clear focus on *adaptive hypermedia systems* can be observed. This cluster contains authors such as Paul de Bra (his four most frequent conferences are Hypertext, WebNet, AH, and EC–TEL), Marcus Specht (EC-TEL, AH, and WebNet), Hugh Davis (ICALT, Hypertext), and Wolfgang Nejdl (AH and many non-TEL conferences focusing on the Web and Information Systems). The cluster also includes personalization as represented in other relevant conferences listed (Judy Kay, for example, publishes most in ITS, AH, and AIED).

Most authors in the *second cluster* have their roots in the field of artificial intelligence – as shown from the main publication venues AAAI and AIED. The conference on Intelligent Tutoring Systems is, in terms of quantity, the most important conference of this cluster. Authors in this cluster include Carolyn Penstein Rose (ITS and AIED), Bruce McLaren (ITS, AIED, and EC-TEL) and Kurt Van Lehn (ITS and AIED). Jim Greer is included in the first two clusters, publishing most in ITS and AIED, but also in EC-TEL and UM conferences, which are closer to the first cluster. Whereas the focus of the first cluster is on personalization and adaptation, the second cluster mainly focuses on understanding learners needs, by applying reasoning techniques to the models of the learner – this can also be observed from the word clouds: “Learn(-er/-ing)”, “Student”, “Model”, and “Cogni(tion)” are the most significant words for this cluster.

The differences in terms of background and focus between the first two clusters are striking, given the similarity in research goals. Learner or user modeling is the first step in the process of adapting a system to the learner [PW05]. It is to be expected that these clusters will become more related with one another, as the targeted conferences AH (first cluster) and UM (second cluster) have merged into the UMAP conference in 2009.

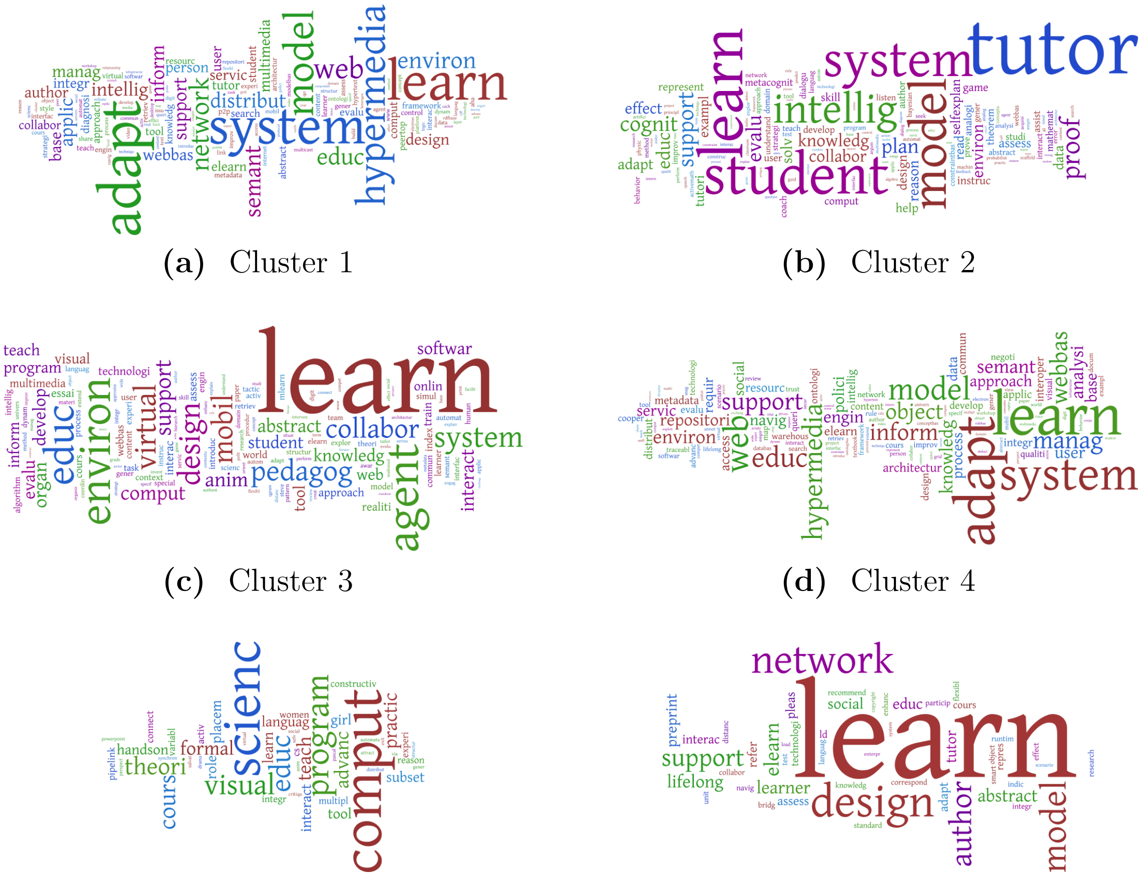
Terms that show up in the *third cluster* are "Environment", "Mobile", "Pedagogy", "Agent", and "Design". Researchers in this cluster have more diverse backgrounds than in the first two clusters, but with the common denominator that they focus on the application of specific technologies to learning. These focuses include mobile technologies (Mike Sharples, Erkki Sukinen – WMTE), computer science education (SIGCSE, Mark Guzdial), and knowledge management.

The *fourth cluster* is an interesting cluster, related to Cluster 1 ("Personalization"), with Peter Brusilovsky as most prominent author. However, this cluster is more focused on learning objects than the first cluster, as witnessed by Erik Duval, as another prominent author. Apart from "Adaptation" and "Hypermedia", the word clouds of this cluster include "Object", "Semantic", "Repository", and "Metadata". As the first cluster, it also includes authors publishing not only in TEL, but in other areas (Ralf Steinmetz and Matthias Jarke). Because of the smaller cluster size, with respect to Cluster 1, several non-TEL related conferences have a bigger impact on the pie chart. That can be considered as an explicit hint as to how other computer science related areas often influence TEL research.

The *fifth cluster* is a very application oriented cluster, with two TEL conferences mostly relating to computer science education (SIGCSE, ITiCSE, Mordechai Ben–Ari as prominent author), and an interesting non–TEL conference on Theoretical Computer Science showing the background of Guido Rling (ENTCS, otherwise publishing mainly in ITiCSE and DeLFI, the German eLearning conference).

In terms of number of publications, Rob Koper is the most prominent researcher in the *sixth cluster*. An online search on these researchers shows that all of them have contributed to the theory of Learning Design [KT05] and related technologies and standards, such as SCORM [SCO] – as exemplified by Baltasar Fernández-Manjón. Not surprisingly, "Learning Design" is the leading term of this cluster's word cloud.

It is apparent that the lists of most popular conferences and journals for each cluster do not only contain TEL-specific conferences: they also contain conferences with a focus on artificial intelligence (AAAI) and human-computer interaction (AH, UM). On the one hand, this shows the importance of these areas to TEL – which matches the numbers of non-TEL venues that we identified during our data collection, as explained earlier in this chapter – but also shows that TEL–related work is presented at other venues. This can be interpreted as evidence for the multidisciplinary character of TEL research. From these six clusters, the *building blocks* of the computer-science related research in TEL can be observed as:



(a) Cluster 1

(b) Cluster 2

(c) Cluster 3

(d) Cluster 4

(e) Cluster 5

(f) Cluster 6

Figure 3.3 A visualization of the TEL research clusters based on paper titles (created using Wordle.net)

- Human–computer interaction, most prominently (adaptive) hypermedia systems (cluster 1);
- Artificial intelligence and (reasoning techniques for) user modeling (cluster 2);
- Semantics, repositories, and metadata (cluster 4).

Cluster 3 and 6 represent the more TEL-specific innovative areas. The terms in their word clouds overlap to a large extent with the "new terms" in ED-MEDIA, as identified by Wild et al. [WVSss]. Finally, cluster 5 presents concrete applications in computer science education.

3.5 Conclusions

In this work, we used author co-citation analysis to analyze and visualize research communities in the area of technology-enhanced learning, focusing on publicly available citation information provided through CiteseerX and conference information available through DBLP. The results are visualized based on relevant conferences and themes for each cluster, providing a first important step to offer a structured overview over research in technology-enhanced learning and make TEL researchers aware of the different research communities relevant for their work.

As an important next step, we will extend our dataset with additional publication and citation data relevant for TEL, most importantly education and psychology, as relevant, for example, for computer supported collaborative learning [5]. These steps are currently performed, together with other project partners, in the context of the STELLAR Network of Excellence.

We hope, that this work as well as future work building on it, will help overcome TEL research fragmentation, by making TEL researchers aware of the different research communities relevant for technology-enhanced learning, and thus more able to bridge communities wherever needed.

[5] The CSCL conference, for example, is not indexed in DBLP and therefore missing in our analysis.

4 Retrospective Event Detection in an Unsupervised Manner

[1]

Content analysis and clustering of natural language documents/articles become crucial in various domains. The goal of this research is to introduce an approach to cluster articles in an unsupervised manner. Clustering documents serves to extract events, where an event is defined as a specific thing happening at a specific time and place, which may be consecutively reported by many articles in a period under observation. Our method is part of the Retrospective Event Detection area (RED). RED is defined as the discovery of previously happened events in historical corpus. The methodology we present can be used as a baseline for seeking any anomalies and for creating a predictive model for the near future in several application domains.

We prove the goodness of our theoretical study adapting our unsupervised learner to the medical domain and in doing that, we suggest a technique which combines aspects from different feature-based event detection methods. We evaluate our approach with two real world datasets with respect to the quality of article clusters. Our results show that we are able to achieve a precision of 60% and a recall of 71% analyzed using manually annotated real-world data.

[1] Image under Creative Commons License avaliable at http://www.flickr.com/photos/digital_dreams/3917888272/sizes/m/in/photostream/

4.1 Introduction

Retrospective Event Detection (RED) is part of the clustering algorithms which have the task to discover previously unidentified events in historical collection. Although RED has been studied for many years, it is yet an open problem.

In this research, we propose a new retrospective event detection algorithm in order to build a predictive model of events for the near future. Our intuition comes from the observation that articles contain two kinds of information: contents and timestamps. The usefulness of time information is often ignored, or at least time information is used in unsatisfied manners. According to these observations, we explore RED and consider the better representations of news articles and events, which should effectively model both the contents and the time information. Also, we notice that previous works consider little on modeling events in probabilistic manners. As a result, in this study we propose a probabilistic model for RED, in which both contents and time information are utilized.

Through this chapter, we will present an application scenario of our methodology. The specific and important domain of health boosted us to apply our approach to real needs in the medical area. In particular, the detected events we extract are defined as Public Health Event (PHE). Actually, a PHE is intended to be some emerging infection, symptom, or illness affecting people or animals in a particular geographic place during a specific time period.

The strategy for epidemiologists to mitigate the impact of potential diseases-spreading is to detect the medical event as early as possible. The body of work devoted to this effort is known as event-based Epidemic Intelligence (EI) [PCKC06]. In order to provide information as timely as possible, by now all the stages of event-based EI system, including document collection, filtering and processing, are done with little or no human intervention. Unstructured and informal text of Web documents are used as data source to detect facts about current infectious disease activity within a population [ea09]. Existing event-based EI systems rely upon the enumeration of possible types of medical reporting patterns (e.g., Medisys [Yan06]). This presents a huge limitation, since given the variety of natural language, many patterns may be required, and the recall for identifying relevant events can be low. Other systems rely on pre-defined keywords for identifying relevant information about public health events. In both cases, the algorithms are not robust enough for the task of detecting emerging public health events, since the only threat indicators (e.g., keywords) they can detect are those that are explicitly under surveillance.

One way to overcome the aforementioned limitations is to cast a new light on the task of public health event detection - so that it is done in an unsupervised and retrospective manner. In this area, there are two major approaches: those based on predictive event detection which mine representative features, given an event [YPC98, LWLM05]; and those that detect an event from a list of highly correlated feature-bursts [FYYL05, HCL07]. The advantages of the former are that they are based on a generative model and they have been shown to provide a more unified framework. They are capable of incorporating multiple modalities such as timestamps as well as explicit representations for various types of entities (e.g., locations, persons, etc.). A drawback is, however, that no prior burst analysis is done on these representations. In contrast, approaches based on the correlation of feature-bursts can filter out a vast number of potentially irrelevant features, yet many types of features relevant for public health event detection (i.e., symptoms, victims, or medical conditions) are not modeled.

We hypothesize that applying an unsupervised algorithm to public health event detection will help to overcome the limitations of existing EI systems. To justify this hypothesis, we propose a new unsupervised approach for event detection. Then, we adapt that to the medical domain in a way that events can be detected that are rare (aperiodic), reoccurring (periodic), and domain or task-specific. In more detail, we combine burst function spectral analysis with entity-centric feature representation of documents in a generative model for predictive event detection. Going beyond a random initialization of the probabilities in this generative process, we instead exploit a known distribution of the features that are obtained directly from the burst function. Additionally, in our burst analysis, we refine the approach to feature representation by incorporating a Cauchy-Lorentz distribution to more closely model the true behavior of periodic, non-burst (trough) activity of features.

4.1.1 Our contributions

The contributions of this work are:

- Use of an approach to unsupervised event detection and adaption of its feature set to the domain of public health event detection.

- Presentation of a general model which, in contrast to previous approaches, incorporates two main techniques: the burst function spectral analysis and the entity-centric feature representation of documents in a generative model. With respect to existing solutions, that results in a more efficient and accurate method to predict public health events.

- Refining the model for representing periodic, non-burst features with the Cauchy-Lorentz distribution. The better sampling reached by such distribution, is shown to be more efficient with respect to the previous representation done by Gaussian distribution [HCL07].

The remainder of this chapter is organized as follows: we discuss about related works in Section 4.2. In Section 4.3, we present details of our generic approach, while in Section 4.4 we characterize the nature of event detection in the public health domain, to lay the foundation for describing the task-specific adaptations required in this setting. Then, in Section 4.5 we expose the experimental results for our approach. Finally, we provide our conclusions in Section 4.6.

4.2 Related Work

The problem of event detection has been examined using news articles as part of a broader initiative named topic detection and tracking [APL98]. The holy grail in this body of work has been to automatically acquire a landscape view of a document collection, which answers in a compact manner the questions of: “What Happened?” and “What is New?”. The governing motivation behind such research was to provide a core technology for a system that would monitor broadcast news and alert an analyst to new and interesting events happening around the world. In Section 4.2.1, we present related works in the area.

Also, the event detection task can be divided into two categories: retrospective event detection (RED) and new event detection (NED), also called “First Story Detection”, in either on-line or off-line mode [YPC98]. Retrospective event detection refers to the detection of previously unidentified events from an accumulated historical collection. New event detection refers to the discovery of the onset of new events, either from live feeds in real-time (online model) or under a closed-world assumption. In particular, the new event detection task is defined to be the task of identifying new events in a set of stories. Each story is processed in sequence, and a decision is made whether or not a new event is discussed. A decision is made after each story is processed [APL98, All02, BCF03, ZZW07]. In this work, we choose to apply a retrospective event detection algorithm using data historical collection, in order to build a predictive model of events for the near future. In Section 4.2.2, we report works in the state of the art.

In general, two main approaches have been considered to solve the problem of event detection, namely: document-based [APL98, YPC98, LWLM05] or feature-based [FYYL05, HCL07]. In document-based approaches, event detection is done by clustering documents (also named articles) based on semantics and timestamps using a generative model of documents. In Section 4.2.2, we show recent works in this direction. In feature-based approaches the temporal and document distributions of words are first studied and events are discovered using distributions of the features over the time, namely trajectory. In Section 4.2.3, we report studies on feature-based approaches focusing also on their temporal behavior.

In contrast to feature trajectory, generative modeling has been shown to provide a more unified framework incorporating multiple modalities such as timestamp of an article and its content. In the generative model, events are latent variables and articles are observations. Latent variables, as opposed to observable variables, are not directly observed, but are rather inferred through the model from some representation of the article’s content that is observable and directly measured.

Through this chapter, we will present an application scenario of our methodology devoted to the public health event detection (i.e. a specific infection, disease, or death happening at a specific time and place) using medical articles. The body of work dedicated to this effort is known as event-based Epidemic Intelligence. In order to address the requirements of Epidemic Intelligence, we incorporate aspects from both the trajectory and generative model approaches. We refine the generative model for features described in [LWLM05] by modeling the features using trajectory distributions that have been computed from the dataset [HCL07]. In Section 4.2.4, we present the major works and systems in the Epidemic Intelligence area.

4.2.1 Topic Detection and Tracking

The Topic Detection and Tracking (TDT) research has provided a standard platform for addressing event-based organization of broadcast news and evaluating such systems [All02]. The governing motivation behind such research was to provide a core technology for a system that would monitor broadcast news and alert an analyst to new and interesting events happening around the world. The research program of TDT focuses on five tasks: story segmentation, first story detection, cluster detection, tracking, and story link detection [LAD+02]. Each is viewed as a component technology whose solution will help address the broader problem of event-based news organization. The details of each of these tasks are beyond the scope of this work. We instead want to focus on the general characteristics of TDT.

To appreciate the unique nature of TDT, it is important to understand the notion of a topic. In TDT, a topic is defined to be a set of news stories/articles that are strongly related by some seminal real-world event. For instance, when an outbreak of enterohemorrhagic Escherichia coli (EHEC) occurred in northern Germany in May and June 2011, it became the seminal event that triggered a new topic. The stories that discussed the origin of this epidemic disease, the consequences on people health, the economical repercussions on the market, etc. were all parts of the original topic. Stories on another epidemic disease (occurring in the same region or time) could make up another topic.

This shows an important contrast with typical Information Retrieval (IR). Along with EHEC in northern Germany, a query "EHEC" will bring up the articles also about other "EHEC" related events. On the other hand, for the query "EHEC in northern Germany", some of the articles that followed the original event in the late spring 2011 might not be considered as about *EHEC in northern Germany* by the traditional IR measures and would be ranked very low in the retrieval set.

This contrast indicates that the notion of an event-based topic is narrower than a subject–based topic; it is built upon its triggering event. Hereafter, we focus on dealing with an event-based topic rather than a subject-based. A typical topic, namely also event, would have a start time and it would fade off from the articles at some point in time. Since it is a specific event, it happened not only at some particular time, but in a specific location, and usually with an identifiable set of participants. In other words, a topic is well defined in scope and possibly in time. In this chapter, we will present experiments on Publich Health Event, paying particular attention to the outbreak of EHEC in northern Germany (see Section 4.5.8).

4.2.2 Retrospective Event Detection

Retrospective Event Detection (RED) is defined as the discovery of previously unidentified events in historical news corpus. It is important to note that some researchers use very similar algorithms to perform both New Event Detection (NED) and Retrospective Event Detection. Thus, some previous work hereafter listed can be categorized into NED as well.

RED was firstly presented and defined by Yang et al. [YPC98], and an agglomerative clustering algorithm (augmented Group Average Clustering, GAC) was proposed; in that paper, articles are represented using their content. From the aspect of utilizing the contents, *TF-IDF* is still the dominant technique for document representation, and cosine similarity is the generally used similarity metric. However, many modifications have been proposed in recent years. Some works focus on finding new distance metrics, such as the Hellinger distance metric [BCF03]. But more works focus on finding better representations of documents, i.e. feature selection. In [YZCJ02], the authors classified documents into different categories, and then removed stop words with respect to the statistics within each category. Significant improvements were reported by them. The usage of named entities have been studied, such as in [LMWY01, YZCJ02]. In [YZCJ02], the authors proposed to re-weight both named entities and non-named terms with respect to statistics within each category. Furthermore, the work presented in [KA04] reports a summarization of the works in this direction and proposed some extensions. They leveraged on using both text classification and named entities to improve the performance. In their work, stop words are removed conditioned on categories, similar with the method in [YZCJ02], but they relaxed the constraint on document comparison. Each document was represented by three vectors: the whole terms, named entities and non-named entity terms. But there are no consistently best representations of documents for all categories.

From the aspect of utilizing time information, generally, there are two kinds of usages. Some approaches, such as the on-line nearest neighbor approach, only use the chronological order of documents. The other approaches, such as [YPC98] and [BCF03], use decaying functions to modify the similarity metrics of the contents.

More recent works such as [LWLM05] and [FSDN10] apply RED using two kinds of information contained within articles: contents, as previously, and timestamps. Authors of both papers prove the importance and usefulness of them in detecting events, going beyond the focus of previous works on finding only better utilizations of contents. In conclusion, the authors propose and improve a probabilistic model for RED, in which both contents and time information are used.

In this thesis, we improve the probabilistic model for RED described in the two aforementioned papers [LWLM05, FSDN10] by combining the model with a deep analysis on the contents using burst function spectral analysis.

4.2.3 Feature based Approaches for Event Detection

In this section, we report researches on feature-based approaches.

One of the latest work on that is paper [FIT11]. It describes a machine learning-based methodology for building an application that is capable of identifying and disseminating healthcare information using as features generic noun-phrases, verb-phrases, biomedical concepts, and bag of words representative techniques. The method extracts sentences from published medical papers that mention diseases and treatments, and identifies semantic relations that exist between diseases and treatments. In detail, the authors work with sentences which they apply different entity extraction techniques and tools to, such as UMLS Metamap as we do (refer to Section 4.5.2), with the purpose to extract highly medical domain-specific concepts.The paper focuses on two main tasks: the first is to identify and classify with a two-class classifier if a sentence is informative or not; the second task is to automatically identify which sentences contain information for three pre-defined semantic relations: *Cure*, *Prevent*, and *Side Effect*.

Previous works focus also on the feautures' temporal behavior. The work in [HCL07] considers the problem of analyzing word trajectories in both time and frequency domains, with the specific goal of identifying important and less-reported, periodic and a-periodic words. The problem of analyzing feature trajectories for event detection uses a well known technique in signal processing to identify distribution of all features by spectral analysis. A set of words with identical trends can be grouped together to reconstruct an event in a completely unsupervised manner. In Section 4.3.2, we go deeper presenting the details of such analysis.

Moreover, referring to the previous Section, so far most TDT research has been concerned with clustering/classifying documents into topic types, identifying novel sentences [AWB03] for new events, etc. without much regard to analyzing the word trajectory with respect to time. In [SA00], the authors attempted to construct an event using co-occurring terms. However, they only considered named entities and noun phrase pairs, without considering their periodicities. On the contrary, our research considers all of the above.

Recently, there has been significant interest in modeling an event in text streams as a "burst of activities" by incorporating temporal information. Kleinberg's seminal work described how bursts for features can be extracted from text streams using an infinite automaton model [Kle02], which inspired a whole series of applications such summarization of evolutionary themes in text streams [MZ05], clustering of text streams using features [HCLZ07], etc. Nevertheless, none of the existing works specifically identified features for events, except for [FYYL05], where relevant features are clustered together to identify events.

Finally, spectral analysis techniques have previously been used in [Vla04] to identify periodicities and bursts from query logs. The authors' focus was on detecting multiple periodicities from the power spectrum graph, which were then used to index words for "query-by-burst" search. In our study, we use spectral analysis to classify word features along two dimensions, namely periodicity and power spectrum (for details see Section 4.3.2). These features with their dimensions are later input to the generative model for detecting event, as shown in Figure 4.1.

4.2.4 Event-based Epidemic Intelligence

Numerous systems exist to detect public health events, notably, PULS [SFvdG$^+$08] uses information extraction technology to identify the disease and the location of a reported event. PULS is integrated into the Medical Information System, Medisys [SFvdG$^+$08, LSF$^+$10], which is a fully automatic public health surveillance system to monitor reporting infectious diseases, chemical, biological, radiological and nuclear threats, food and feed contaminations, and plant health. Medisys automatically gathers reports concerning Public Health in various languages from many Internet sources world-wide, classifies them according to hundreds of categories, detects trends across categories and languages, and notifies users. Furthermore, the system categorizes all incoming articles according to pre-defined multilingual categories, identifies entities (e.g., diseases and locations), clusters news articles, and calculates statistics to detect emerging threats, called *Alerts*. We report a description about how Medisys computes and detects alerts in Section 4.5.9. Furthermore, users can screen the categorized articles which can be further filtered by language, news source, and country. Articles are classified in a category, if they satisfy the category definition.

Proteus-BIO [GHY02] automatically extracts infectious disease outbreak information from several sources including ProMed-mail [Mad04], the World Health Organization (WHO) [Org], and medical news web sites. EpiSpider [KBTea09] extracts publication dates, locations, and topics from ProMed-mail reports and news, converting them to a semantic interchange format suitable for storage in a relational database.

BioCaster [CDK$^+$08] is an ontology-based system which uses a domain-specific event ontology to perform named entity recognition on outbreak reports. The system analyzes documents reported from over 1,700 RSS web feeds[2], classifies them for topical relevance, and plots them onto a Google map using geo-code information. BioCaster's Text Mining process is based on rules that are capable of matching a number of elements including entity classes, skip-words, entity types, or regular expression patterns.

In general, the advantages of event extraction based on text mining process and

[2] http://en.wikipedia.org/wiki/RSS

information extraction are that they are: deterministic, produce clear granularity results (sentence level), produce explicit and well structured unit of information, and the outputs are easily interpretable. Further, event extraction allows a system to specifically tune learners to detect specific type of information and capture linguistics and semantic relations.

In contrast to these systems, we propose an approach that exploits unsupervised machine learning technology. None of the existing methodologies consider the use of such approach. We believe that an unsupervised approach can complement existing systems since it allows to identify public health events (PHE), even if no matching keywords or linguistic patterns can be found. To corroborate our idea, we show comparison results between Medisys and our approach UPHED in Section 4.5.8 and in Section 4.5.9.

4.3 Unsupervised Event Detection

An event is defined as a specific episode happening at a specific time and place [Dp], which may be consecutively reported by many articles in a period.

The goal of this work is to introduce an approach to detect events in an unsupervised manner. The model can also be used as a baseline for detecting any anomalies and for building a predictive model for the near future.

We consider a three stages process for detecting events (Figure 4.1):

1. In the first stage, **Named Entity Feature Representation**, we build entity-centric document surrogates. The manner in which we extract features and represent documents is outlined in Section 4.3.1.

2. We then perform **Feature Analysis** on the extracted set of features to prune the less relevant ones. Details are reported in Section 4.3.2.

3. The resulting set of features is then used as input for the **Unsupervised Event Detection** stage. Section 4.3.3 spells out how the detection is conducted.

Finally, a fourth future step can be identifying relations between detected events which later can be aggregated.

4.3.1 Named Entity Feature Representation

As first step, we process raw text to build an entity-centric feature representation of each document. Given a collection of text documents, we define a finite set of articles $\mathcal{A}$, as well as an Event Template $\mathcal{T}$. The template $\mathcal{T}$ represents a set of feature types, which are important for describing events. More specifically, we can describe an event by several attributes that provide information, e.g., on *who* is involved by

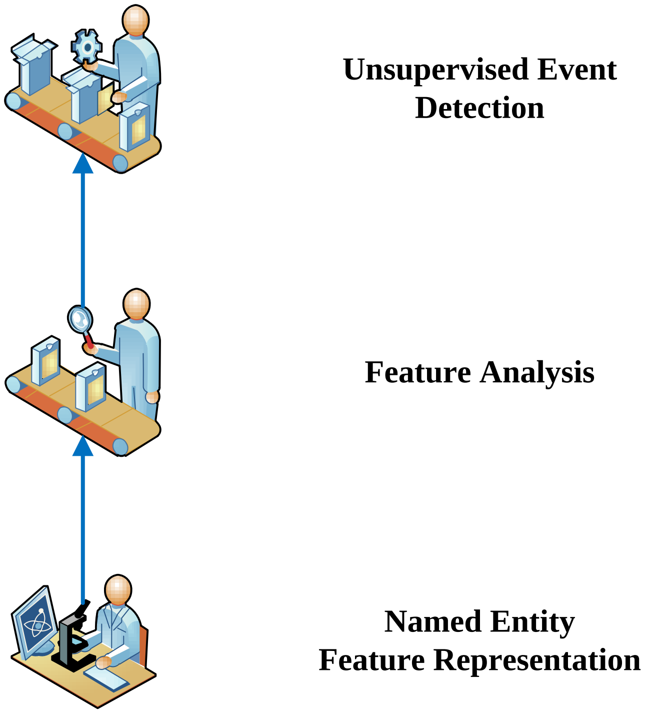

Unsupervised Event Detection
Feature Analysis
Named Entity Feature Representation

Figure 4.1 Overview of Unsupervised Event Detection

what, *where* (locations), and *when* (time, defined as the period between the first relevant article and the last relevant one reporting the event). Thus, the template is instantiated as:

$$\mathcal{T} =< Entity^*, Time > .$$

Instances of the template elements are represented as $< entity^*, t >$, where $entity^*$ refers to several entity types like *persons* (*who* is involved) and *locations* (*where* event is happening). With t we refer to *time* (*when* event is happening). With such a template, we focus on the utilization of the contents and time information of articles and we model them with different models. On the one hand, articles are always aroused by events; on the other hand, similar articles reporting the same event often redundantly appear on many sources. The former hints a generative model of news articles, and the latter provides data enriched environments to perform event detection [LWLM05]. With consideration of these characteristics, we propose a probabilistic model to incorporate both content and time information in a unified framework. This model gives new representations of both articles and events as specified hereafter.

Content: The content of each article is represented by a bag-of-words whose type is given by the event template. For each document, a vector is created for each of the feature types; each entry in the vector corresponds to the frequency with which

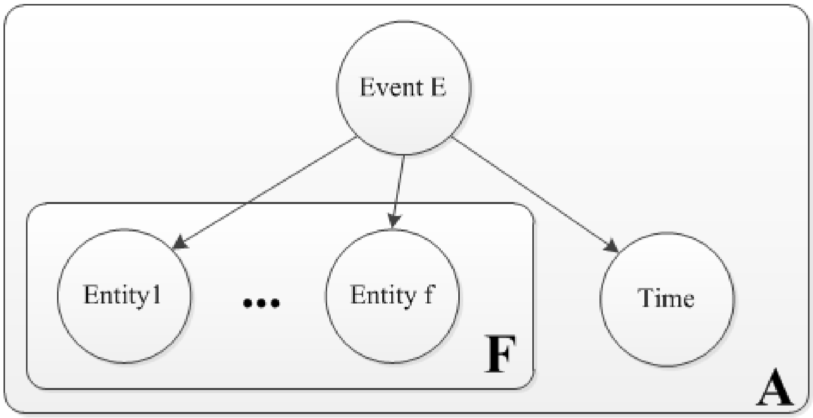


Figure 4.2 Graphical model representation

a feature, i.e. an entity of a given type, appears in the bag-of-words representation.

Time: Each event corresponds to a peak on an article count versus time distribution. In other words, the distribution is a mixture of many distributions of events. A peak is usually modeled by a Gaussian function, where the mean is the position of the peak and the variance is the event's duration. As a result, a $GaussianMixtureModel$ is chosen to model times.

An article also can be represented by *entity** and time (represented by a discrete value, i.e. the timestamp). As a result, we describe an ***Event*** and an ***Article*** using the following tuples:

$$Event =< \{entity\}^*, time(Period) >$$

$$Article =< \{entity\}^*, time(Timestamp) >$$

Figure 4.2 is a graphical representation of this model, where F is the term space size of all kinds of entities (e.g., in Figure 4.2 the count of all kinds of entities is f). Furthermore, within the figure the concept/node Event E is on top of the other nodes, since in the unsupervised event detection an Event is a latent variable, whose value is defined with respect to the observed content of articles, i.e. *Entities* and *Time*, by a generative process. A indicates the article set.

In order to simplify our model, we assume that all kinds of information of the i-th article, given an event e_j, are conditional independent. Thus, the probability of an article i to be associated to an event e_j is given by the product of the following individual probabilities:

$$p(article_i|e_j) = \left(\prod_{f=1}^{F} p(entity_{fi}|e_j)\right) * p(time_i|e_j)$$

Where f represents the kind of entity, e.g., location type, person type, etc.

4.3.2 Feature Analysis

In our work, we posit that event detection involves a dual task: the detection of periodic as well as aperiodic events. With respect to a window of one year, for example, aperiodic events are also important, since they can represent an event that is annual, e.g., season flu, or quite severe and life threatening, such as a sudden outbreak of EHEC.

Detection of periodic as well as aperiodic events is based on identification of periodic and aperiodic features as described in [HCL07] using a common technique such as the spectral analysis. In this approach, features are classified with respect to their periodicity (P_w) and their dominant power spectrum (S_w).

The periodicity of a feature refers to its frequency of appearances. If the feature is *aperiodic*, then it occurs once within the period P, and its P_w has a value equal to the period itself. If the feature is *periodic*, then it happens regularly with a fixed periodicity, i.e. $P_w \leq \lceil P/2 \rceil$. The periodicity is a function of the dominant power spectrum which is computed via the discrete Fourier transform applied to the feature distributions, for more details refer to the following paragraphs.

Representative Features

Intuitively, an event can be described very concisely by a few discriminative and representative word features and vice-versa, e.g., "ehec" and "northern Germany" could be representative features of an outbreak of enterohemorrhagic Escherichia coli (EHEC) occurred in northern Germany.

Let P be the duration/period (in days) of a collection of articles, and F represents the complete word feature space. The representation vector of a word feature is defined as follows:

Definition 1 ***Feature Trajectory:*** *The trajectory of a word feature w can be written as the sequence*

$$y_w = [y_w(1), y_w(2), \ldots, y_w(P)]$$

where each element $y_w(t)$ is a measure of feature w at time t, which could be defined using the normalized $DF - IDF$ score

$$y_w(t) = \frac{DF_w(t)}{N(t)} * \log\left(\frac{N}{DF_w}\right)$$

Spectral Analysis for Dominant Period

Given a feature w, we decompose its feature trajectory $y_w = [y_w(1), y_w(2), \ldots, y_w(P)]$ into the sequence of P complex numbers $[X_1, \ldots, X_P]$ via the discrete Fourier transform DFT:

$$X_k = \sum_{t=1}^{P} y_w(t) * e^{-\frac{2\pi i}{P}(k-1)t},\ k = 1, 2, \ldots, P$$

DFT can represent the original time series as a linear combination of complex sinusoids, which is illustrated by the inverse discrete Fourier transform $IDFT$:

$$y_w(t) = \frac{1}{P}\sum_{k=1}^{P} X_k * e^{-\frac{2\pi i}{P}(k-1)t},\ k = 1, 2, \ldots, P$$

where the Fourier coefficient X_k denotes the amplitude of the sinusoid with frequency k/P.

The original trajectory can be reconstructed with just the dominant frequencies, which can be determined from the power spectrum using the popular periodogram estimator. The *periodogram* is a sequence of the squared magnitude of the Fourier coefficients, $||X_k||^2$, $k = 1, 2, \ldots, P/2$, which indicates the signal power at frequency k/P in the spectrum.

From the power spectrum, the dominant period is chosen as the inverse of the frequency with respect to the highest power spectrum, as follows.

Definition 2 ***Dominant Period:*** *The dominant period of a given feature w is*

$$P_w = \frac{P}{\arg\max_k ||X_k||^2}$$

Accordingly, we have

Definition 3 ***Dominant Power Spectrum:*** *The dominant power spectrum of a given feature w is*

$$S_w = ||X_k||^2,\ with ||X_k||^2 \geq ||X_j||^2,\ \forall j \neq k$$

In conclusion, the dominant power spectrum, S_w, of a feature w is a strong indicator of its activeness at the specified frequency; the higher is the S_w, the more likely the feature is to be relevant within the dataset. Thus, S_w can be considered to filter out irrelevant features, i.e. features with a dominant power spectrum less than a prefixed threshold chosen according to the domain. After filtering out irrelevant features, the remaining features are meaningful and could potentially be representative for some events [HCL07].

Identifying Burst for Aperiodic Features

Let $y_w(t)$ be the distribution of the feature w over the time t under observation; further, let $y_w(t)$ be computed as exposed in Definition 1 (refer also to [HCL07]).

Then, for each *aperiodic* feature, we keep only the bursty period which is modeled by a Gaussian distribution.

$$f_{ap}(y_w(t)) = \frac{1}{\sqrt{2\pi\sigma_w^2}} * e^{-\frac{1}{2\sigma_w^2}(y_w(t)-\mu_w)^2} \tag{4.1}$$

The well known Expectation Maximization (EM) algorithm is used to compute the Gaussian density parameters μ_k and σ_k [DLR77].

Identifying Bursts for Periodic Features

To model the periodic features we chose a mixture of K Cauchy-Lorentz distributions, where $K = \lfloor P/P_w \rfloor$. Such a distribution is similar to the Gaussian, but differs in the thickness of its tails. This property, as observed from the computed $y_w(t)$, reflects better the distribution of *periodic* features, since, even for t far from the peak of the burst (non burst or trough activity), generally the feature distribution $y_w(t)$ reports a value that is important to be considered. The mixture is described as follows:

$$f_p(y_w(t)) = \sum_{k=1}^{K} \alpha_k * \frac{1}{\pi}\left[\frac{\gamma}{(y_w(t) - \mu_w) + \gamma^2}\right] \tag{4.2}$$

for the mixture proportions α_k of assigning y_w into the k^{th} Cauchy-Lorentz distribution.

$$0 \leq \alpha_k \leq 1 \text{ where } \sum_{k=1}^{K} \alpha_k = 1, \forall k \in [1, K] \subset \mathbb{N} \tag{4.3}$$

Furthermore, μ_w is the location parameter, specifying where is the peak of the distribution, and γ is the scale parameter which specifies the half-width at half-maximum. μ, γ and α are computed using the EM algorithm [DLR77].

Feature burst distributions algorithm

In this section, we present the algorithm for computing the feature burst distributions. Algorithm 5 wraps together the concepts and the approaches explained so far. The novelty of this approach is to represent periodic features with the Cauchy-Lorentz distribution. Finally, using the feature burst distributions for having a more representative model has been proved to be successful in Section 4.5.

4.3.3 Detecting Events

A core step, in the unsupervised detection of events is the clustering of articles and generation of events. Formally, from this stage we get the following sets of conditional probabilities that are shown in Table 4.1. We use these probabilities, as a basis for determining that an event has occurred.

Algorithm 5: Feature Analysis using feature burst distributions

```
Input: A set of extracted features $W$; a set of articles $A$; a fixed threshold $\tau$
Output: all the feature burst distributions
begin
    $N$ := count the number of articles within $A$;
    $D$ := count the number of distinct date $t$, according to articles' timestamps, in $A$;
    $P[|W|]$ := array storing the dominant period $P_w$ for each feature $w$;
    $S[|W|]$ := array storing the dominant power spectrum $S_w$ for each feature $w$;
    $FeatureDistributions[|W|][D]$ := Matrix storing the vectors of feature distributions $y_w$ for each feature $w$ over the dates $t$;
    $FourierFeatureDistributions[|W|][D]$ := Matrix storing the decomposition of the vectors of feature distributions $y_w$ into the sequence of complex vectors via the discrete Fourier transform $DFT$;
    for each distinct date $t$ in $A$ do
        $N(t)$ := count the number of documents at date $t$;
    for each feature $w$ in $W$ do
        $DF_w$ := count the total number of documents containing entity $w$;
        for each distinct date $t$ in $A$ do
            $DF_w(t)$ := count the number of documents containing feature $w$ at date $t$;
            DF-IDF := $\frac{DF_w(t)}{N(t)} * \log\left(\frac{N}{DF_w}\right)$;
            Store DF-IDF into $FeatureDistributions[w][t]$;
    Compute $FourierFeatureDistributions$ using $DFT$ on $FeatureDistributions$;
    for each feature $w$ in $W$ do
        $P[w]$ := compute the dominant period $P_w$ of the corresponding feature;
        $S[w]$ := compute the dominant power spectrum $S_w$ of the corresponding feature;
        if $S[w] \geq \tau$ then
            if $P[w] > \left\lceil \frac{P}{2} \right\rceil$ then
                model the feature burst by a Gaussian distribution (aperiodic feature);
            else
                model the feature burst by a mixture of $K = \lfloor P/P_w \rfloor$ Cauchy-Lorentz distributions (periodic feature);
end
```

Probability	*Description*
$P(a\|e)$:	Set of conditional probabilities for an article a, given an event e
$P(w\|e)$:	Set of conditional probabilities for a feature w, given an event e
$P(e)$:	Set of probabilities for an event e

Table 4.1 Probabilities obtained from unsupervised event detection

Approach

Numerous techniques exist for detecting events in an unsupervised way (see Section 4.2). As previously mentioned, in this work we choose to apply a retrospective event detection algorithm using data historical collection, in order to build a predictive model of events for the near future. Additionally, we choose a probabilistic generative model for event detection, because it has been proven to be a more unified framework for handling the multiple modalities (i.e. time, content, entity types) of an article and its content.

Our unsupervised event detection algorithm is based on the Retrospective Event Detection (RED) algorithm presented by Li et al. [LWLM05]. It relies on a generative model where the articles are produced using multinomial distributions over the feature types. These articles are used later as starting points for a clustering relying on the iterative EM algorithm [DLR77]. In addition, in their work, the multinomial distributions are initialized with random probabilities. Thus, the generated articles are randomly picked.

As part of our approach, we refine the RED algorithm by going beyond this random initialization of probabilities – exploiting the feature distributions from our Feature Analysis stage (Section 4.3.2). The underlying intuition for our approach is based on proven results, which show that an initial starting point estimated in a better-than-random way can, in fact, be expected to speed up the iterative EM algorithm converging closer to the optimum log-lokelihood (introduced later in this sub-section) of a collection of articles, than an initial point that is picked at random. For more details refer to [ZZH$^+$09]. In our approach, we aggregate the computed feature distributions over the articles, and use this information into the multinomial distributions of the generative model. Thus, the generated articles, used as starting points by the EM algorithm, are not totally randomly picked.

Although it has been proven that retrieved events are not influenced by the starting points [Hof99, SG07], the EM algorithm needs to be restarted several times with several different random starting points in order to get a good approximation of events. Supported by the analysis in [ZZH$^+$09], we do not need multiple restarts of the EM algorithm, since an initial starting point estimated in this way can be expected to be closer to the optimum than a randomly picked initial point.

For sake of clarity, here we highlight what introduced in Section 4.3.1. Events in the unsupervised event detection are latent variables, whose value is defined with respect to the observed content of articles by a generative process. An event is defined by a pattern of entity-centric features that co-occur with some saliency and probability from the observable content of the articles which contain mentions of these features. Since the set of articles that describe the same event contain similar sets of feature co-occurances, the articles themselves cluster and are described by the conditional probability of an article, given an event $p(a_i|e_j)$.

Generative Model for Events

Our generative model is described in Algorithm 6.

Algorithm 6: Detection of Public Health Events: the generative model

```
begin
    Choose an event $e_j \sim \text{Multinomial}(\theta_j)$;
    Generate an article $a_i \sim p(a_i|e_j)$;
    Draw a time stamp $time_i \sim N(\mu_j, \sigma_j)$;
    for each feature of $a_i$, according to the type $k$ of current feature do
        Choose an $entity_{fi} \sim \text{Multinomial}(\theta_f^j|time_i)$;
end
```

In the algorithm, the vector θ_j represents *event* probabilities initially instantiated randomly (here the definition of *event* is according to the formalization of the multinomial distribution); μ_j and σ_j are parameters of the conditional Gaussian distribution given event e_j; θ_f^j is a vector of probabilities computed by aggregating the feature burst distributions over the $time_i$ of a given event e_j.

Learning Generative Model Parameters

The model parameters can be estimated by Maximum Likelihood method following the approach described in [LWLM05]. By introducing latent variable, i.e. events, we can write the log-likelihood of the joint distribution as:

$$l(X;\theta) \propto \log(p(X|\theta)) = \log\left(\prod_{i=1}^{A} p(a_i|\theta)\right) = \sum_{i=1}^{A} \log\left(\sum_{j=1}^{K} p(e_j)p(a_i|e_j,\theta)\right) \qquad (4.4)$$

where X represents the corpus of articles; A and K are the number of articles and the number of events respectively. As presented in Section 4.3.1, given an event e_j, all kinds of information of the i-th article are conditional independent:

$$p(a_i|e_j) = \left(\prod_{f=1}^{F} p(entity_{fi}|e_j)\right) * p(time_i|e_j) \tag{4.5}$$

where f represents the kind of entity, e.g., location type, person type, etc.

Expectation Maximization (EM) algorithm is generally applied to maximize log-likelihood. The parameters could be estimated by running E-step and M-step alternatively.

In E-step, we compute the posteriors, $p(e_j|a_i)$, by:

$$P(e_j|a_i)^{(t+1)} = \frac{p(e_j)^{(t)} p(a_i|e_j)^{(t)}}{p(a_i)^{(t)}} \propto p(e_j)^{(t)} p(a_i|e_j)^{(t)} \tag{4.6}$$

where the upper script (t) indicates the t-th iteration. In M-step, we update the parameters of each mixture model. Since entities, in the space F with reference to Figure 4.2, are modeled with independent mixture of unigram models, so their update equations are the same, and we use token f_n to represent the $entity_{fn}$.

For the mixture of unigram models, parameters are updated by:

$$P(f_n|e_j)^{(t+1)} = \frac{1 + \sum_{i=1}^{A} p(e_j|a_i)^{(t+1)} * tf(i,n)}{N + \sum_{i=1}^{A} \left(p(e_j|a_i)^{(t+1)} * \sum_{s=1}^{N} tf(i,s)\right)} \tag{4.7}$$

where $tf(i,n)$ is the count of entity f_n in a_i and N is the vocabulary size. For each type of entities, N is the size of corresponding term space. Since the co-occurrence matrix is very sparse, we apply Laplace smoothing [NMTM00] to prevent zero probabilities for infrequently occurring entities in Equation 4.7.

The parameters of the *Gaussian Mixture Model* are updated by:

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{A} p(e_j|a_i)^{(t+1)} * time_i}{\sum_{i=1}^{A} p(e_j|a_i)^{(t+1)}} \tag{4.8}$$

and

$$\sigma_j^{(t+1)} = \frac{\sum_{i=1}^{A} p(e_j|a_i)^{(t+1)} * (time_i - \mu_j^{(t+1)})^2}{\sum_{i=1}^{A} p(e_j|a_i)^{(t+1)}} \tag{4.9}$$

It is important to note that because both the means and variances of the Gaussian functions change consistently with the whole model, the Gaussian functions work like sliding windows on a time line. In this way, we overcome the shortcomings caused by the fixed windows or the fixed decaying function parameters used in traditional news event detection algorithms [BCF03]. At last, the mixture proportions are updated by:

$$P(e_j)^{(t+1)} = \frac{\sum_{i=1}^{A} p(e_j|a_i)^{(t+1)}}{A} \tag{4.10}$$

The EM algorithm increases the log-likelihood consistently, while it will stop at a local maximum.

4.4 Application Scenario: Public Health Event

Content analysis and clustering of natural language documents become crucial in public health. Recent pandemics such as Swine Flu have caused concern for public health officials. Given the ever increasing pace at which infectious diseases can spread globally, officials must be prepared to react sooner and with greater epidemic intelligence gathering capabilities than before. Events such as emerging infectious diseases, are those considered to be either completely new or reoccurring. An important strategy used by officials to mitigate the impact of potential threats, is to find ways to detect the signs of a public health event as early as possible.

To address that, in this section, we propose to apply our approach to detect public health events (PHE) in an unsupervised manner. We address the problems associated with adapting our method of an unsupervised learner to the medical domain and in doing so, hereafter we propose an approach which combines aspects from different feature-based event detection methods. Later in Section 4.5, we evaluate our approach with two real world datasets with respect to the quality of article-clusters.

Formally, a public health event is defined as a specific infection, disease, or death that happens at a specific time and place, which may be consecutively reported by many medical articles in a period.

In a typical epidemic investigation task, public health officials must detect anomalous behaviors. They periodically compute statistics about disease reporting events, using the recent past, in order to build a predictive model for the near future. The model is used as a baseline for detecting any anomalies. These statistics are based on aggregated information which, in our case, is derived from detecting events in an unsupervised manner from documents.

We consider a five stages process for detecting unsupervised public health events, adapting what discussed in Section 4.3 and presented in Figure 4.1 to this scenario:

1. In the first stage, **Named Entity Feature Representation**, we build entity-centric document surrogates that are suitable for the medical domain. The manner in which we extract features and represent documents is outlined in Section 4.4.1.

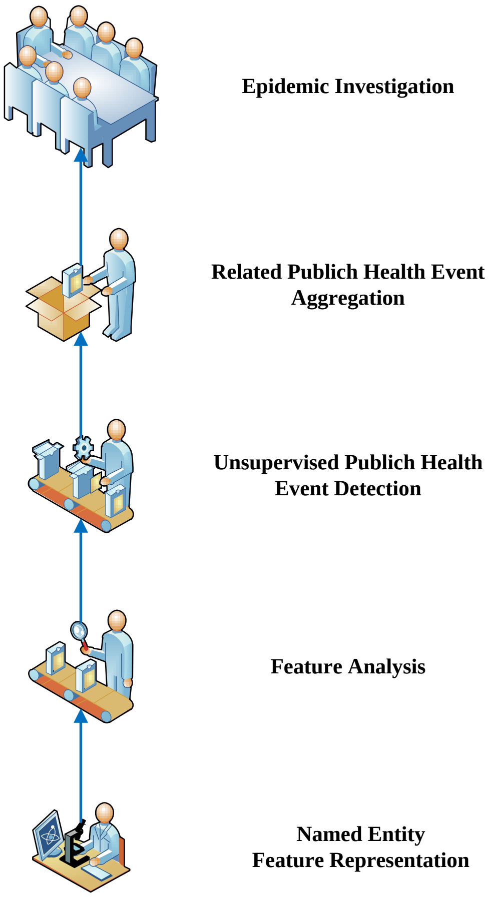

Epidemic Investigation
Related Publich Health Event Aggregation
Unsupervised Publich Health Event Detection
Feature Analysis
Named Entity Feature Representation

Figure 4.3 Overview of Unsupervised Public Health Event Detection

2. We then perform **Feature Analysis** on the extracted set of features to prune the less relevant ones. Details are reported in Section 4.4.2.

3. The resulting set of features is then used as input for the **Unsupervised Public Health Event Detection** stage. Section 4.4.3 spells out how the detection is conducted.

Finally, to perform at the end an **Epidemic Investigation** where officials detect anomalous behaviors, relations between detected events need to be aggregated within the **Related PHE Aggregation** phase. The latter task will be considered in future work, while the previous ones can be already applied letting epidemiologists investigate the public health events extracted.

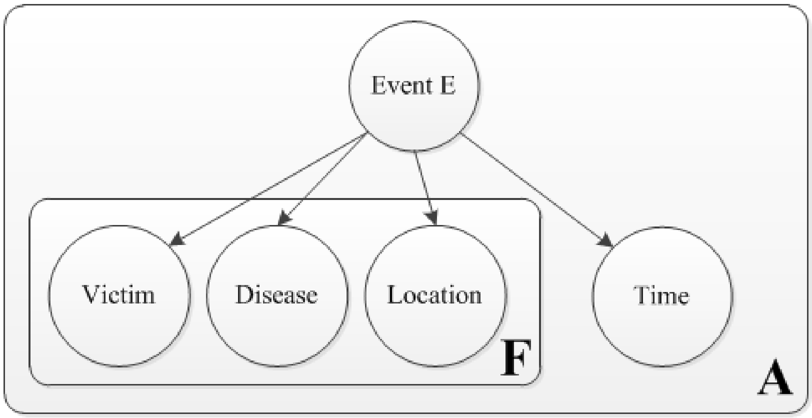


Figure 4.4 Graphical model representation for the medical case

4.4.1 Named Entity Feature Representation

According to Section 4.3.1, here we discuss how to represent named entity features for our medical scenario.

Given a collection of text documents, we define a finite set of medical articles $\mathcal{A}$, as well as a Health Event Template $\mathcal{T}$. The template $\mathcal{T}$ represents a set of feature types which are important for describing public health events. More specifically, we describe a public health event by four attributes that provide information on *who* (victims) was infected by *what* (diseases), *where* (locations), and *when* (as before, time is defined as the period between the first relevant article and the last relevant one in this domain). Thus, the template is instantiated as:

$$\mathcal{T} =< Victim, Disease, Location, Time > .$$

Instances of the template elements are represented as $< v, d, l, t >$ for victim, disease, location, and time, respectively. Content and time are modeled according to our approach presented in Section 4.3.1.

As a result, we describe a PHE and a medical article using the following tuples:

$$PHE = \{victims, diseases, locations, time(Period)\}$$

$$article = \{victims, diseases, locations, time(Timestamp)\}$$

Figure 4.4 is a graphical representation of this model for the medical case, where F is the term space size of the three kinds of entities (i.e. Victim, Disease, and Location). A indicates the article set.

In order to simplify our model, we assume the four kinds of information of the i-th medical article, given a PHE e_j, are conditional independent. Thus, the probability of an article $article_i$ to be associated to an event e_j is given by the product of the

Probability	*Description*
$P(a\|e)$:	Set of conditional probabilities for a medical article a, given an event e
$P(v\|e)$:	Set of conditional probabilities for a *victim* v, given an event e
$P(d\|e)$:	Set of conditional probabilities for a *disease* d, given an event e
$P(l\|e)$:	Set of conditional probabilities for a *location* l, given an event e
$P(e)$:	Set of probabilities for an event e

Table 4.2 Probabilities obtained from unsupervised PHE detection

following individual probabilities:

$$\begin{aligned} p(article_i|e_j) =& p(victims_i|e_j)p(diseases_i|e_j)p(locations_i|e_j) \\ & p(time_i|e_j) \end{aligned}$$

4.4.2 Feature Analysis

Epidemic Intelligence focuses on the detection of periodic as well as aperiodic events. In our domain, periodic events need to be determined in order to build statistical models for reoccurring infectious diseases, or to track the changes in the prevalence level for an outbreak once it has occurred. Aperiodic events as well show new outbreaks and are important since they can give epidemiologists information about new trends in diseases-spreading. Given the nature of public health event detection, it is important to be able to model both types of events for EI.

This stage uses Spectral analysis as a common technique for identifying periodic and aperiodic features, as described in Section 4.3.2.

4.4.3 Detecting Public Health Events

For sake of clarity, applying to the medical case what treated in Section 4.3.3, in this stage we get the following sets of conditional probabilities that are shown in Table 4.2.

Approach

In this work, we choose to apply a retrospective event detection algorithm since it is important in EI to use a collection of historical data, in order to build a predictive model of public health events for the near future. The same idea is adopted in statistical methods used in public health to analyze event data from indicator-based systems, such as the Farrington Algorithm [CNAM96]. This algorithm compares the current number of cases to a threshold computed from historical data. Farringtons approach is used when the pathogen incidence varies over months/seasons, but is also quite stable between years.

Additionally, we have chosen a probabilistic generative model for public health event detection, as introduced in Section 4.3.3.

Generative Model for Public Health Events

Our generative model is described in Algorithm 7 and it is a domain adaptation of Algorithm 6.

Algorithm 7: Detection of Public Health Events: the generative model

```
begin
    Choose an event $e_j \sim$ Multinomial($\theta_j$);
    Generate a medical article $a_i \sim p(a_i|e_j)$;
    Draw a time stamp $time_i \sim N(\mu_j, \sigma_j)$;
    for each feature of $a_i$, according to the type of current feature do
        Choose a $victim_{iv} \sim$ Multinomial($\theta_v^j|time_i$);
        Choose a $disease_{id} \sim$ Multinomial($\theta_d^j|time_i$);
        Choose a $location_{il} \sim$ Multinomial($\theta_l^j|time_i$);
end
```

With respect to Algorithm 6, here we introduce an observation on θ_v^j, θ_d^j, θ_l^j which are vectors of probabilities computed by aggregating the feature burst distributions for *victims*, *diseases*, and *locations* over the $time_i$ of a given event e_j. In other words, for example, let $victim_{iv}$ be the v-th instance of *Victim* for the generated article a_i; let e_j be an event; thus, we initialize the probability $P(victim_{iv}|e_j)$ according to the previously computed distribution of the v-th instance of *Victim* over the time.

4.5 Experiments and Evaluations

In order to analyze the results of the introduced method, namely the Unsupervised Public Health Event Detection (*UPHED*), we ran several experiments. For the specific task considered here, which is public health event detection, no annotated data set is available. Anyway, we performed some analysis on a real-world data set. In this section, the data set used for the experiments is introduced together with the experimental settings and results.

Finally, in section 4.5.8, we run our method on a different data set than the one presented in section 4.5.1, proving its effectiveness detecting a recent outbreak of enterohemorrhagic Escherichia coli (EHEC) occurred in northern Germany starting in May 2011.

4.5.1 Dataset

To build our data set, we collected source documents from the *url* column of the PULS online fact base [SFvdG+08], a state-of-the-art event-based system for Epidemic Intelligence which provides public health event summarization and search capabilities. The data were collected for a four months period, from September 1 to December 31, 2009, by crawling the website. In total $1,397$ documents were collected. The data were processed by stripping all boilerplate and markup codes using the method introduced by Kohlschütter et al. [KFN10].

The reduced data set size is due to the unavailability of standard evaluation data sets for public health event detection, thus a ground truth was manually built as explained in Section 4.5.5. Nevertheless, our data set size is comparable with other data sets reported in relevant works [Kaw10, LWLM05, MWC10, NAXC08].

4.5.2 Feature Set

In the experiments, the algorithm is run on a feature set consisting of named entities. Table 4.3 presents the main categories of features collected and their counts. The entities have been extracted using two different named entity recognition tools: UMLS MetaMap [Met] and OpenCalais [Ope].

OpenCalais was used to recognize *medical conditions* and all variants of *location*. MetaMap was used to identify the *victim* features. MetaMap has originally been developed for indexing biomedical literature and relies upon the Unified Medical Language System (UMLS) Metathesaurus, a very rich biomedical vocabulary designed and mantained by the US National Library of Medicine. Thus, it allows extracting highly domain-specific concepts, but leads when applied to social media or news articles to false positives. For our feature set we are only interested in disease names and symptoms which are more reliable detected by OpenCalais. In contrast, the more detailed information on victims provided by MetaMap is very useful for our algorithm. For these reasons, we decided to exploit these two different named entity recognition tools.

Through manual inspection, we further found that noise introduced into the algorithm due to multi-word expressions causes an explosion of the number of features. This is particularly acute for a feature-centric approach such as ours in the medical domain, in which features, consisting of many multi-word expressions, quite commonly exacerbate the problem of producing irrelevant events. For this reason, we

Feature Types	**Feature Categories**	*norm*	*unnorm*
Victims	Population Group Age Group Family Group Animal	28	4100
Diseases	Medical Condition	917	2754
Locations	City ProvinceOrState Country Continent	955	982

Table 4.3 Overview on the collected features. *norm* is the number of normalized features; *unnorm* is the total number features before the normalization process

normalized the features by the use of a taxonomy. More specifically, the numerous and more specific concepts that are lower in the taxonomy are re-represented by a parent concept. Because of that, we rely upon the taxonomy underlying the UMLS semantic network. As an example, the terms *boy, girl, baby, child, kid* were normalized to the single feature, *child*.

4.5.3 Experiment I: Feature Pruning

Objectives: The features described before were analyzed by computing for each feature the periodicity P_w, and their dominant power spectrum S_w. The objective of this analysis was to prune the less important features, i.e. those that are potentially not representative to some event. The pruning was driven by looking at the S_w since it represents the relevance of features within the dataset.

As claimed in [HCL07], setting a threshold over S_w, to identify which features to discard, is more of an art than science and it is domain specific. We further provide some examples of periodic and aperiodic features either pruned or kept.

Results: In Figure 4.5, we report the values of the dominant periods P_w, over the dominant power spectrums S_w, for each feature extracted (indicated as a point in the diagram). In the graph one can notice an empty area between $P_w = 62$ and $P_w = 123$. According to the definition of periodicity, the bounds of this empty area correspond respectively to $\lceil P/2 \rceil$ and P, where P is the period under observation. Looking at the distributions of all the *feature* points over the graph, we chose to set the threshold τ over S_w to 0.5 (τ is specified in Algorithm 5). In Figure 4.5, the pruned features are those points to the left of the vertical line set at $S_w = 0.5$.

Finally, in Table 4.4, we present some selected features, categorized by periodicity

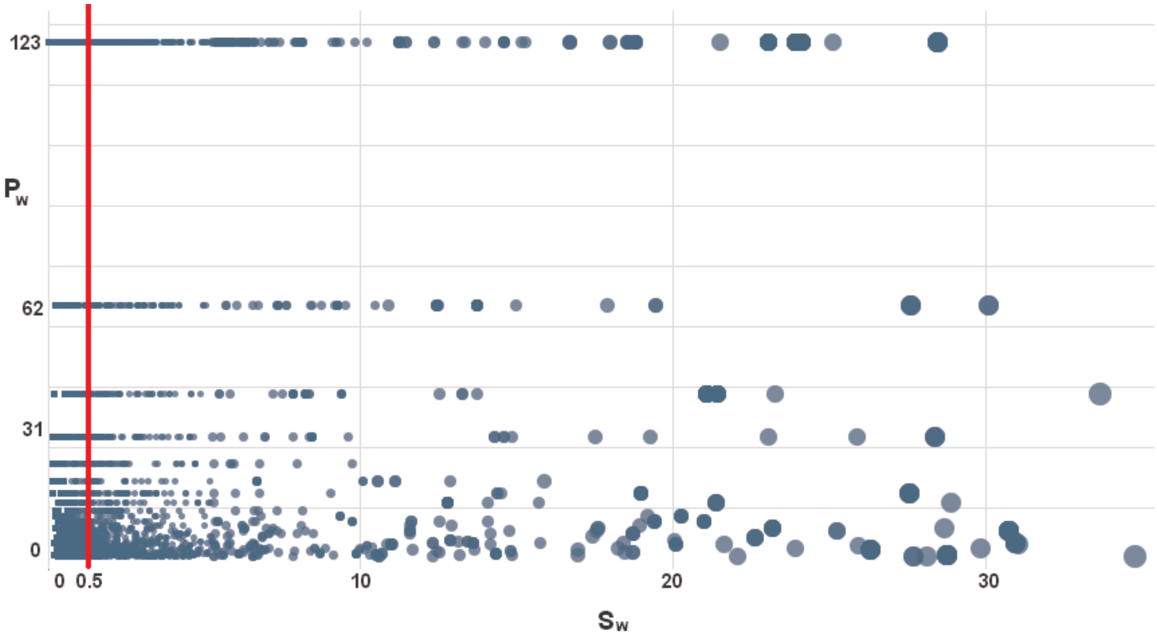


Figure 4.5 Dominant periodP_w and dominant power spectrum S_w of all extracted features

Features	Pruned	NOT pruned
Aperiodic	Kuru epidemic, Kashan, filariasis, leprosy, light disease, Minnesota, Nottingham, Slovenia, Yangon, etc.	cholera, Japan, leptospirosis, Manila, SARS, spanish flu, Surrey, syndrome CFS, Ukraine, United Kingdom, etc.
Periodic	Arizona, autumn disease, season disease, coughing, Karachi, laugh disease, Mali, Nevada, pneumonia, travel infection, weekly infection, etc.	Africa, disorder, fever, flu, infection, HIV, H1N1 (influenza), parkinson, rabies, tetanus, etc.

Table 4.4 Features categorized by periodicity and relevance

k	10	**15**	20	25	50	100
F1-measure	0.35	**0.68**	0.33	0.34	0.33	0.41

Table 4.5 F1-measure over different number of events k

and relevance. The features on the first column are not relevant since they have a value of S_w less than τ and are pruned, while the second column collects the relevant features.

On the surface, the pruned terms seem quite relevant, but manual inspection revealed that they actually represent single instance terms within the dataset. Further, since terms that did not conform to the Health Event Template remain unconsidered, all features (both pruned and non-pruned) express some domain-specific association.

4.5.4 Experiment II: Selection of k

Objectives: A key consideration in a retrospective event detection is the determination of the number of events k to be used as input for the generative model. The choice of the number of events can affect the interpretability of the results. For example, a solution with too few events will generally result in very broad events. A solution with too many events will result in un-interpretable events that pick out idiosyncratic feature combinations. We used a hill-climbing approach to discover the number of events as done in [LWLM05]. This method detects all peaks on the articles count-time distribution, and then computes salient scores for each of them. A proportion of the number of salient peaks is then used as an initial estimation of the number of events; such a proportion is an experimentally determined parameter. The objective of this experiment is to determine the value of k which the algorithm performs best for.

Results: We ran our method ten times, each time setting a different input value for the number of events k. Table 4.5 presents the *F1-measure* values computed for different number of events. It can be seen that the best response from our approach was achieved with $k = 15$, which corresponds to the 50% of all peaks on the articles count-time distribution. Based on this result, we chose $k = 15$ for the manual assessment presented in the Experiments III and IV.

4.5.5 Experiment III: Cluster Quality

Objectives: An underlying intuition in our approach is that an unsupervised event detection algorithm, which is tuned to detect public health events, would produce better quality clusters with respect to the Epidemic Intelligence task, than either a public health event extraction system based on information extraction, or a generic event detection algorithm.

In order to verify this assumption, we evaluated the quality of three clustering techniques. For the first clustering, *PULS* was used standing for a public health event extraction system based on information extraction. For the second clustering, the approach described in [LWLM05] was adopted as a generic event detection algorithm. The latter clustering considers time, locations, persons, as well as general keywords as features in the algorithms; we use *RED* to identify it. Finally, the third clustering was created using our approach, namely the Unsupervised Public Health Event Detection (*UPHED*).

Due to the unavailability of standard evaluation data sets for public health event detection, a ground truth was manually built. To construct the ground truth, each document was manually examined by three subjects, who were instructed to assign each document to one of the 15 clusters. Each document examined contained only a subset of the sentences - specifically, those containing at least one of the relevant named entities of medical condition, location, and person. Each named entity was highlighted in the sentence. Discrepancies among the subjects were resolved by a fourth subject, based on mutual agreement. The manually built clusters were then used as ground truth to evaluate the precision, recall, and f1-measure for the set of clusters that were created by the event detection algorithm.

Results: Table 4.6 shows the results for precision (P), recall (R), and f1-measure (F). As it can be seen, *UPHED* performs better than the other two ones; thus, incorporating the medical conditions as entity type in fact and integrating the burst function analysis in the generative model, it helps to improve the clustering performance with respect to *RED*. Moreover, it has to be noticed that the RED technique drops with reference to the results shown in [LWLM05]; this can be justified stating that the RED works well in general domain, but worse when adopting a particular domain as the Epidemic Intelligence.

	P	R	F
PULS	40%	31%	34.9%
RED	40.6%	51.5%	45.4%
UPHED	**59.7%**	**70.6%**	**64.7%**

Table 4.6 Results for three clustering approaches

Regarding the clustering from *PULS*, although the achieved results seem relatively low, an important difference should be noted. There is a difference in vocabulary (feature set) due to the named entity extraction process used in PULS and our work. Although we sought to use the same number of features – as presented in the PULS fact base – the distributions we obtained for the features still differ. For example, the locations used in PULS are exclusively at the granularity of the country, whereas those extracted from the corresponding raw PULS text (for our work), contain multiple geographical levels, such as cities, providences, states, etc. Second, for the medical conditions we encountered a similar context: terms caught by UPHED (e.g., coughing, vomiting, disorder, etc.) were not present in the PULS fact base, since they focused on actual disease mentions.

We believe that accounting for the difference in the feature set would improve PULS results. However, it is important to remind that we propose an approach that exploits unsupervised machine learning technology, and PULS does not consider the use of such approach, but it is based only on matching keywords or predefined linguistic patterns. Furthermore, PULS can be complemented by our method since our algorithm allows to identify public health events (PHE) even if no matching keywords or linguistic patterns can be found.

Additionally, a standardized data set for the field would also allow for a stronger comparison of the approaches.

Finally, we noticed that the magnitude of our results does not exceed a value of 71% for recall. An explanation is that manually assigning documents to clusters is a very difficult task. Other sources of data containing labels to validate clustering results will be considered in the future. Possibly, one source could be the WHO's update headings which are entries that group related outbreak reports together.

4.5.6 Experiment IV: Detected Public Health Events

Objectives: In another experiment, we manually analyzed the clusters together with the documents assigned to them; thus, we tried to describe the underlying events using natural language terms or even finding an official information source reporting the event. Furthermore, we separated periodic from aperiodic events. The objective of this analysis was to provide a human interpretation of the clusters and to have a critical analysis on them. The separation between aperiodic and periodic events provides information about the earlier re-occurring of some medical event within our time window.

Results: Table 4.7 shows the characterizing cluster terms resulted from our clustering for a value of k of 15. The *Event Terms* shown in this table were presented to the testers in the manual assessment described in the section before. For each feature type, the two most probable features have been selected for being shown. For each article we evaluated its conditional probability given an event, i.e. $P(a|e)$ presented in Table 4.1. We imposed a threshold $\tau_p = 5\%$ for each $P(a|e)$; whereas the conditional probability under observation was upper than τ_p, then the document was associated to the event (cluster), otherwise it was discarded. The number of articles assigned to the single clusters is shown in Table 4.7. For sake of clarity, according to the constraint introduced with the threshold τ_p, the total number of articles associated to events is 1254, lower than the number of all documents within the dataset, i.e. 1397.

Event Id	Event Terms	No. Docs	Event Description
E_1	wales, united kingdom, swine flu, flu, people, children	43	In November 2009, there were a couple of new swine flu cases occurring in Wales.
E_2	china, beijing, flu, swine flu, people, female	93	In October 2009, China had an increased number of swine flu cases. Further, the population was vaccinated to a large extent.
E_3	New York, united states, flu, swine flu, children, people	144	In November 2009, the flu death toll increased significantly in U.S.
E_4	bangalore, india, flu, infection, people, children	60	In September 2009, the swine flu toll in Bangalore increased.
E_5	japan, tokyo, disease, swine flu, people, children	65	In October 2009, Japan started with swine flu vaccinations.
E_6	france, europe, disorder, flu, people, female	58	In September 2009, an increased number of swine flu cases were reported in France.
E_7	surrey, london, e.coli, diarrhea, children, animals	79	In September 2009, there was an outbreak of E. coli in England. Mainly children were concerned who visited the same farm in Surrey.
E_8	manila, malaysia, disease, infection, people, father	117	In October 2009 there was an outbreak of leptospirosis in Manila, the capital of Philippines. Further, there was a typhoon which led to an increase of several infectious diseases which interested also the close Malaysia.
E_9	africa, Kenya, cholera, diarrhea, female, people	98	In December 2009 a deadly outbreak of cholera in northwestern Kenya took place.
E_{10}	delhi, united states, flu, dengue, children, people	93	In November 2009 there was an outbreak of Dengue in Delhi. Contemporaneously, in United States several outbreaks of swine flu happened.
E_{11}	china, asia, disease, rabies, animals, people	54	In September 2009, China's health officials became aware that rabies had become one of the biggest public health risks facing China.
E_{12}	united states, finland, cancer, chronic fatigue syndrome, people, animals	49	An american researcher found out that a virus linked to prostate cancer may also be linked to Chronic Fatigue Syndrome. The same research was conducted by a finnish study (October 2009).
E_{13}	nigeria, south africa, disease, infection, children, people	81	Several studies found out that babies and children in Africa die from infections (September 2009). Further, there was a measles campaign in South Africa (October 2009). **Period**: every 7 days.
E_{14}	united states, canada, vomiting, swine flu, people, children	132	President Obama stated the fight against swine flu (October 2009). Also, several outbreaks of swine flu in U.S. and Canada happened (October and November 2009). **Period**: every 4 days.
E_{15}	united states, russia, swine flu, disease, people, female	88	Several news articles provide comparison of swine flu statistics for various countries, comparing mainly cases happening in Russia and U.S. **Period**: every 4 days.

Table 4.7 Detected aperiodic (E_1 - E_{12}) and periodic events (E_{13} - E_{15}). Columns respectively show extracted terms, number of documents, and brief description of the real events

In addition, the table presents a manually created description of the occurred event that is reflected by most of the documents in the cluster. Evidently, most of them are very relevant events that express, of course, the global event happening in 2009 which is *swine flu*. For the events reporting such a disease, an outbreak was also detected by Google Flu Trends[3]. Nevertheless, some of the discovered events refer to

[3] http://www.google.org/flutrends/

some other diseases.

It can be seen that sometimes one extracted medical entity, used to describe the cluster, is the general term *disease*. This mainly happens when different diseases are described in the single documents. In documents, the disease itself is often mentioned only a few times and often replaced by the more general term *disease*. For this reason, the calculated probability of the term *disease* became higher than other more specific medical conditions; in conclusion, our algorithm chose *disease* as cluster describing term.

Some cluster labels reflect the content of the documents quite well - the terms seem to be consistent with the reported event (e.g., cluster E_1). For other clusters, one might get the impression that two different events are described when looking at the terms. For example, cluster E_{10}, where a subset of the documents assigned to this cluster deals with dengue in Delhi and another subset of documents refers to swine flu cases in United States. It can also be seen that the clusters are somehow overlapping. For example, the event related to *swine flu* in *U.S.* is reflected by at least two clusters, i.e. E_3 and E_{14}.

For some of the detected events we could even find official press releases from health organizations through manual assessment. For example, the event described by cluster E_7, which refers to an outbreak of E.coli in England, can be confirmed by a press release of the Health Protection Agency on September 13, 2009[4]. The terms selected for describing the cluster reflect very well this event: children were infected by E.coli after a visit in a farm in Surrey. Further, we separated and reported aperiodic (E_1 - E_{12}) from periodic events (E_{13} - E_{15}).

In summary, from this manual assessment we learned that documents reporting a similar or the same event are clustered together by the algorithm. The event terms that are selected based on their probability to describe the content of the clusters reflect the content of the documents quite well; they even reflect the case when different events are assigned to the same cluster.

4.5.7 Experiment V: Efficiency Comparison

Objectives: In this section, we compare three strategies for detecting events:

1. The baseline of our method, which initializes the EM algorithm with random points, as done in [LWLM05], and adapted to the medical domain. We use *Rdm* to represent it.

2. An approximation of our method identifying bursts for periodic features using a mixture of $K = \lfloor P/P_w \rfloor$ Gaussians, as suggested in [HCL07]. We use *GaussApp* to identify it.

[4] http://www.hpa.org.uk/NewsCentre/NationalPress Releases/2009 PressReleases/

3. Our proposed method, namely the *UPHED*.

The intention of this analysis is to show that the selection of a good starting point can boost the EM algorithm to converge quickly to the optimum and that it is unnecessary to restart the EM algorithm multiple times with different random starting points, as done previously.

	Rdm	*GaussApp*	*UPHED*
Optimum (log-likelihood)	-3648	-3543	**-3497**
Best Starting Point (log-likelihood)	-3929	-3704	**-3665**
Average running time (seconds)	18.40	12.91	**11.32**
Average number of iterations for *EM*	10	7	**5**
Best trial: number of iterations for *EM*	7	**4**	**4**
Worst Trial: number of iterations for *EM*	14	10	**8**
Average number of restart of *EM* **to get the optimum**	5-6	**1-2**	**1-2**

Table 4.8 Efficiency comparison of three different strategies

Results: All methods, from the three compared strategies, were ran on a machine with an Intel T2500 2GHz processor, 2GB memory under OS Fedora 9. The algorithms were all implemented in Java compiled by JDK 6.0.

The experimental results are shown in Table 4.8. Here, we report the log-likelihood both for the optimum and for the best starting points of the three strategies. The log-likelihood indicates how likely the documents are generated by models, so it is the bigger the better. Then, we relate the average running time and, since this can be affected by implementation preferences, we describe the number of iterations of the *EM* algorithm to converge to the optimum in the best case, in the worst case, and on average. Finally, we show the number of restarts the *EM* algorithm needs for converging to the optimum, since the presence of local maxima can mislead the algorithm to reach the global maximum.

From the results, we can conclude that in all the reported measures, our proposed method *UPHED* performs more efficiently than the other two ones. This also assists the conclusion that *UPHED* is much easier to get to the optimum than starting from a random point, thus needs less time and fewer iterations to converge.

4.5.8 Experiment VI: Effectiveness

Objectives: In this experiment, we study the large outbreak of enterohemorrhagic Escherichia coli (EHEC) occurred in northern Germany in the period from May till

June 2011. Several thousand people denounced symptoms of haemolytic uremic syndrome (HUS) and gastroenteritis. In this section, we illustrate how our approach effectively detected this medical event. Also, our clustering method allowed identifying different aspects within the unfolding medical events, analyzing general media and official sites gathered using Medisys [FWC+11]. Beside the outbreak under consideration, our algorithm detected other medical events occurred during the selected time window, hereafter specified.

To build our data set, we collected source documents from Medisys feed [5], which we applied entities extraction procedure on, as outlined in Section 4.5.2. The data were gathered for a 2 months period, from beginning of May till the end of June 2011. In total 13, 076 sources were stored. For sake of clarity, the same feature set as in Section 4.5.2 was considered with respect to *Diseases* and *Locations*.

Event Id	**Event Terms**	**No. Docs**	**Event Description**
E_1	Germany, LowerSaxony, EHEC, HUS, gastroenteritis	1,723	Outbreak of enterohemorrhagic Escherichia coli (EHEC) occurred in northern Germany.
E_2	Spain, EHEC, haemolityc	957	The contagion was caused by contaminated Spanish vegetables.
E_3	Russia, EHEC	450	Russia applied trade restriction for European vegetable products.
E_4	France, EHEC, gastroenteritis, vomiting	387	Many cases of EHEC contagion in France.
E_5	Sweden, Germany, EHEC, diarrhea, HUS	801	First Swedish tourist group visiting northern Germany denounced EHEC infection.
E_6	Brussels, Luxembourg, EHEC, HUS	439	Documents treating the economical repercussion on the European market of the entire EHEC outbreak.

Table 4.9 Detected EHEC outbreaks during May and June 2011. Columns respectively show extracted terms, number of documents, and brief description of the real events

Results: We run our method on 13, 076 resources. Out of those, 4, 757 were categorized in six medical events regarding EHEC outbreak, as shown in Table 4.9, and 4, 639 were clustered in further six medical events occurred during the beginning of the second quarter 2011, as reported in Table 4.10.

Both tables show the characterizing cluster terms resulted from our clustering. As in Section 4.5.6, for each article we evaluated its conditional probability given an event, i.e. $P(a|e)$ presented in Table 4.1, imposing a threshold $\tau_p = 5\%$ for each $P(a|e)$. In addition, the number of documents assigned to single cluster is shown, as well as a manually created description of the occurred event that is reflected by most of the documents in the cluster. For sake of clarity, according to the constraint introduced with the threshold τ_p, the total number of articles associated to events is 9, 396, lower than the number of all documents, i.e. 13, 076. The explanation has to be found in the following two reasons. The first one is that not classified documents

[5] http://medusa.jrc.it/m-eco?language=en

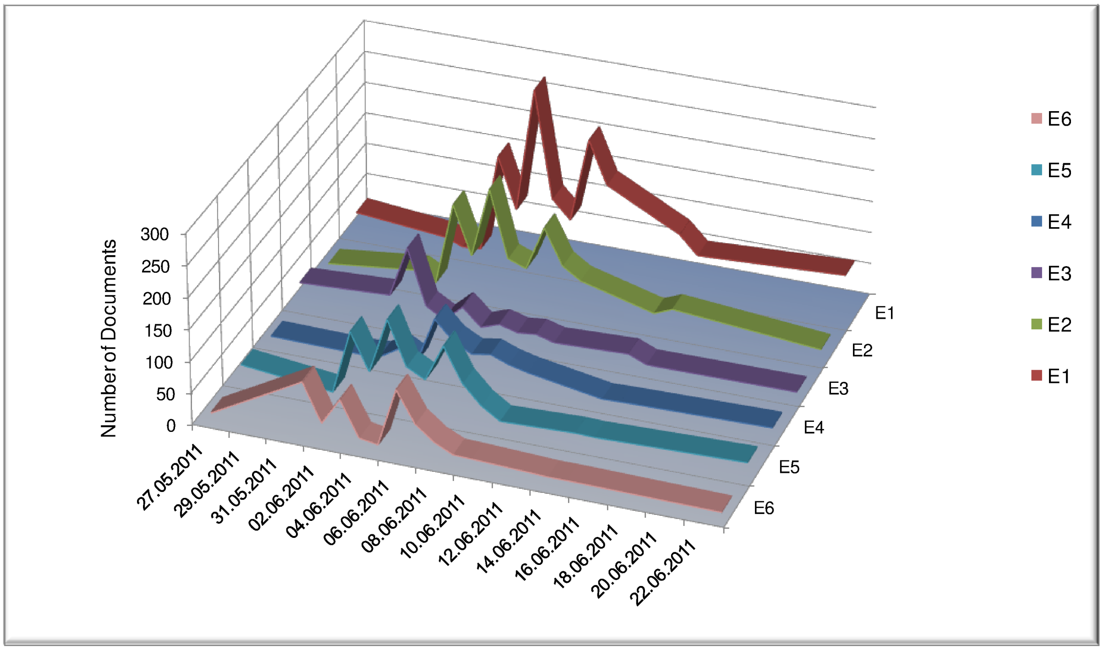


Figure 4.6 Documents distributions for each extracted medical event relevant to EHEC outbreak

are news about celebrities in a context where a disease is mentioned and should be considered unwanted noise. The second one is simply that our algorithm considered such documents irrelevant (e.g., about sports, films, etc.) according to the imposed threshold τ_p. Differently from the experiment in section 4.5.6, the *Event Terms* shown were collected selecting for each feature type (i.e. Diseases and Locations) the most probable features exceeding the probability threshold τ_f of 40% to be successfully associated to such medical event.

All medical events reported in Table 4.9 are relevant to the EHEC outbreak during May and June 2011 and show how the media attention changed geographical focus over time, following the developing situation. Instead, medical events presented in Table 4.10, are related to other public health events detected during the same period.

To better emphasize EHEC scenario, in Figure 4.6 we clearly identify key aspects of each extracted event over time. In details:

1. E_1 presents an outbreak of enterohemorrhagic Escherichia coli (EHEC) occurred in northern Germany, with a sudden rise on articles first detected at the end of May; later, another rise was observed between June 5 and June 11 according to German authorities announcing that bean sprouts were the source of infection.

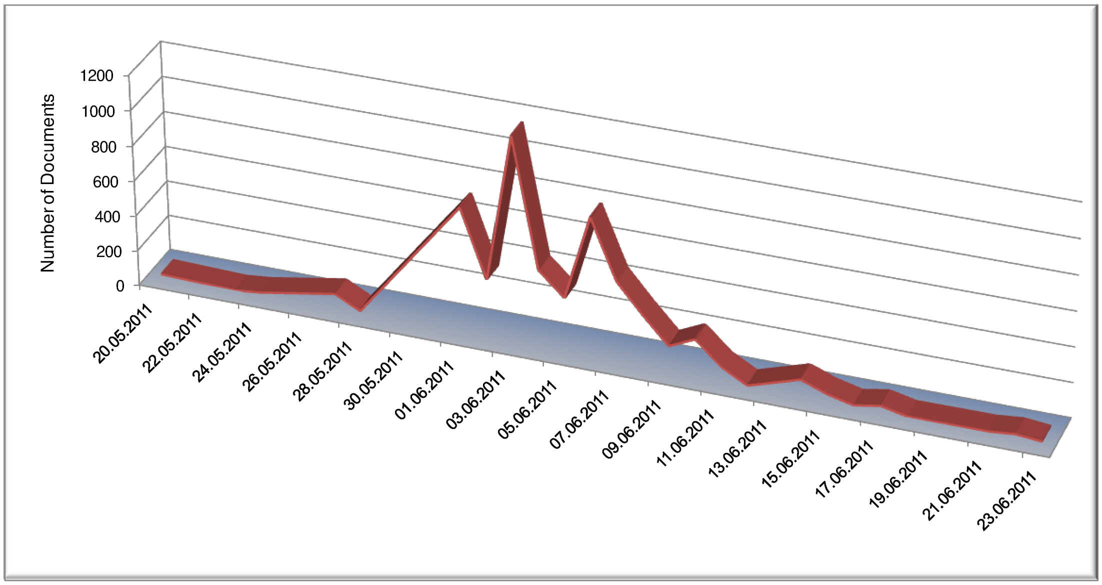


Figure 4.7 Cumulative documents distributions of all events (from E_1 to E_6) for EHEC over time

2. E_2 reports that the contagion was caused by contaminated Spanish vegetables. This statement was alleged at the end of May, as can be seen in the graph with an increasing trend of the curve, while there was another rise on June 1 when Spanish farmers announced that Spanish cucumbers had been tested negative for EHEC.

3. E_3 cites when Russia applied trade restriction for European vegetable products with a peak of documents on June 3.

4. E_4 refers to many cases of EHEC contagion in southern France observed between June 3 and June 7.

5. E_5 reports an increasing trend with articles on May 27, on the first Swedish tourist group visiting northern Germany who denounced EHEC infection.

6. E_6 clusters together all articles mentioning the European Commission during the discussion on the alleged contaminations of Spanish cucumbers and vegetables, with a peak on sources during June 1 and June 3; furthermore, the European Commission and Parliament were involved on themes about the risk

Event Id	Event Terms	No. Docs	Event Description
E_7	Italy, measles, epidemic	322	Outbreak of measles occurred in northern Italy.
E_8	France, measles, rubella, fever	448	The majority of measles cases in Europe arose in France, around 72% of all measles occurrences.
E_9	Europe, measles	2,574	The second quarter 2011 registered a peak of measles cases in all Europe.
E_{10}	India, New Delhi, malaria	571	India's scientific community is all set to launch a research programme on how to better combat vector-borne diseases, after many cases of malaria in the country.
E_{11}	China, Beijing, , tuberculosis, malaria	407	China has been sustaining efforts that have progressively achieved coverage of the country's vast population by tuberculosis treatment and surveillance.
E_{12}	Australia, Melbourne, malaria	317	"UT Southwestern Research Team's Anti-Malarial Work" wins the "International Project Of The Year Award".

Table 4.10 Further detected medical events during May and June 2011

assessment in terms of public health at EU level, with a rise between June 6 and June 9.

For sake of transparency, in Figure 4.7 we show the distribution of articles of all six medical events about EHEC computed over time. As can been noticed, a peak was observed on June 3.

Results on medical events related to the EHEC outbreak in Europe were also caught and presented by the Medisys Report [LMF+11]. Our method complements what presented in the report, since we captured the latent factor in Medisys documents without a manual analysis document by document, as was done in [LMF+11] to reach similar outcomes presented here. Thus, we speed up the process to detect outbreaks, since we can point out directly users or investigators to documents containing important information about potential medical outbreak, better focusing their attention on health needs. Surely, human inspection is always needed in this scenario and cannot be left out, but we can provide a mechanism to use better and better epidemic inspectors efforts.

We continue our analysis describing the details of further 6 events detected by our $UPHED$ technique and reported in Table 4.10 as follows:

1. E_7, E_8, and E_9 refer to a big occurrence of measles cases, around ten thousand reported cases in Europe. Of the total, 72% were detected in France, almost with an incidence of ten times more than measles cases occurred in the same period one year before (2010).

2. E_{10} reports that India's scientific community is ready to launch a research programme which will bring to vaccinate millions of people and save another 6.4 million lives over the current decade.

3. E_{11} clusters together articles mentioning medical improvements in China during the past 20 years through better infectious disease surveillance systems which

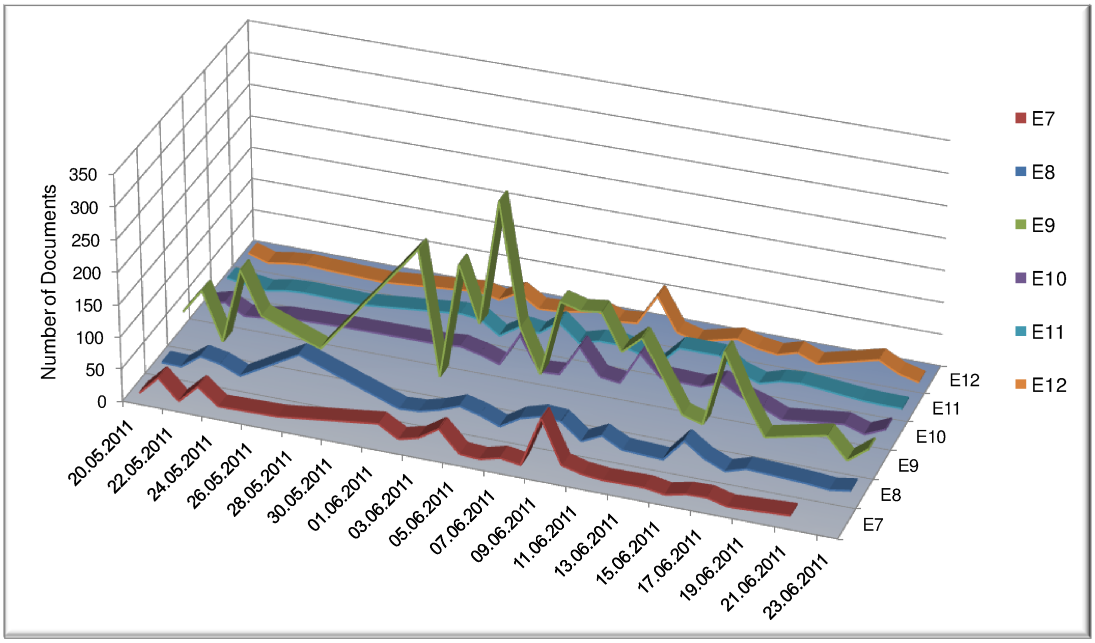


Figure 4.8 Documents distributions for each extracted medical event NOT relevant to EHEC outbreak; from E_7 to E_{12}

have led to decrease the mortality for tuberculosis of 8-6% per year. The success in China has been based on sustained efforts that have progressively achieved coverage of the country's vast population by tuberculosis treatment and surveillance.

4. E_{12} cites when the Australian research team anti-malarial, which began in 2002 under the direction of Dr. Margaret Phillips, identified a promising inhibitor of a specific enzyme that the malaria parasite requires for survival. For that, her research team won the International Project Of The Year Award.

As last analysis, in Figure 4.8 we identify key aspects of each detected event over time.

4.5.9 Experiment VII: *UPHED* in comparison with *Medisys*

Objectives: The experiment conducted hereafter extends Section 4.5.8 since leverages on the dataset and medical events previously extracted. In summary, we provide an extensive comparison of our *UPHED* algorithm with *Medisys*, which is a fully automatic public health surveillance system run by the Health Threats Unit at Directorate General Health and Consumer Affairs of the European Commission, in collaboration with the Joint Research Centre in Ispra, Italy. More details about Medisys were presented in Related Work (Section 4.2) and are available in [SFvdG+08, LSF+10].

The comparison was evaluated analyzing the *alerts* generated by $Medisys$ versus the *events* detected by our $UPHED$ method.

Preliminaries: Medisys allows the selection of articles about any subject via Boolean combinations of search words or lists of search words, organized into classes such as *Countries*, *Communicable Diseases*, *Animal Diseases*, *Organizations*, etc. All the search words are multilingual. Each subject definition is called *alert*, which, according to the nature of the search words, is multilingual [LSF+10].

Due to the high number of independent news sources, Medisys captures many reports that readers of one or a few news sources would miss. The drawback of having multiple resources consists in reporting the same or near duplicate documents to users and triggering much more alerts as in reality. To cope with that, Medisys adopts a similarity measure to prune near-duplicate documents. The similarity measure for the news articles is based on cosine similarity on a simple vector-space representation of the first 200 word tokens of each article. This means that not only multiple reports of the same story, but also similar reports about different cases for the same disease may be grouped together and filtered out. This method allows to discard entire groups of non-influent articles at once.

With the alert definitions in Medisys, mention of a disease, an animal disease, a country, or others can be identified in multiple languages. Medisys keeps a running count of all disease alerts for each country, i.e. it maintains the average of all documents mentioning a specific category instance and country, over a time window of two weeks. The category instance can be a word within the aforementioned classes such as Communicable Diseases, Animal Diseases, Organizations, etc. An alerting function detects a sudden increase in the number of articles for a given category and country, by comparing the statistics for the last day with the two-week rolling average. The more articles there are for a given *category-country* combination compared to the expected number of articles (i.e. the two-week average), the higher is the alert level. The histogram in Figure 4.9 illustrates seemingly how Medisys presents statistics on the alerts on its web site. On the left side of the figure, alerts with high alert level are shown. In particular, red bars identify peaks of documents which triggered high level alerts; blue bars show the average number of documents of the last two weeks for the combination under inspection. As can be noticed,the higher is the deviation of the peak from the 14-day average value, the more important is the alert. Finally, on the right side of the figure, alerts with medium level are reported and represented by yellow bars.

Technically, alert levels are calculated assuming a normal distribution of articles per category over time. Alert levels are high, if the number of articles found is at least three times the standard deviation over the last 14 days, while alert levels

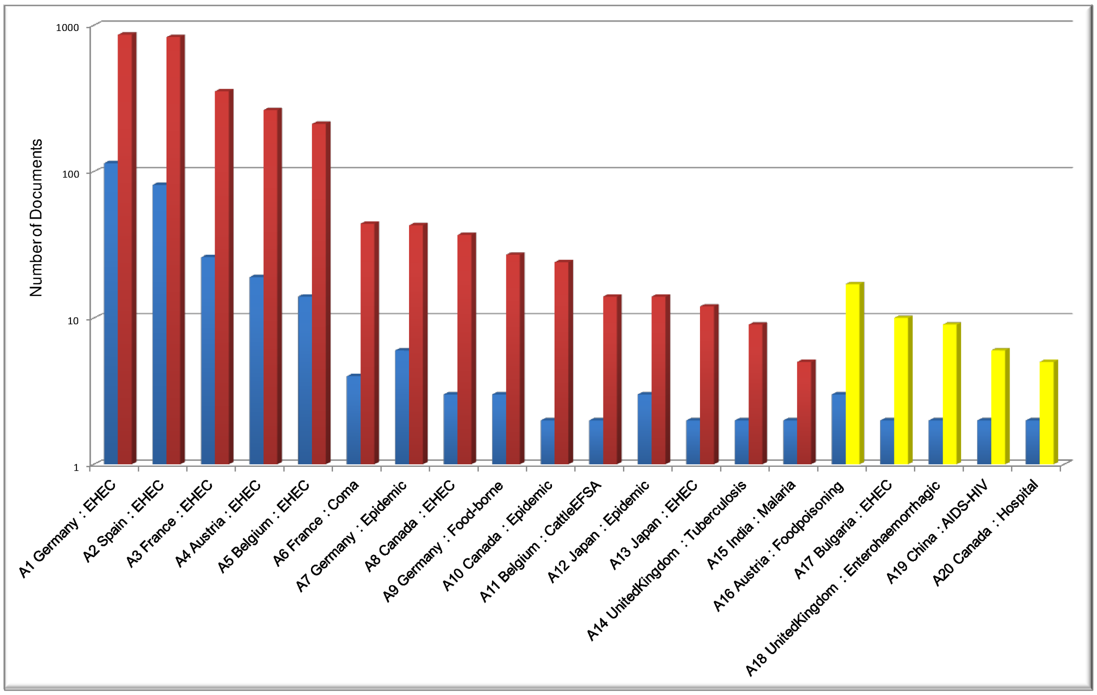


Figure 4.9 Selection of 20 alert statistics from *Medisys* from beginning of May till end of June 2011. Term *EFSA* identifies the European Food Safety Authority

are medium, if the number of articles found is between twice and three times the considered standard deviation. As the total number of articles varies during the week (fewer articles on Sunday and Monday), a correction is applied to the documents' frequencies according to the day of the week [SFvdG^{+}08].

To accomplish our comparison, a RSS tunnel feed [6] has been set up between Medisys and UPHED. At present, UPHED processes only English-language documents. Our method triggers Medisys feeds every minute. Medisys sends through the tunnel documents which it categorizes as relevant to the medical domain. Currently, the documents arrive as plain text and UPHED applies its entity extraction (Section 4.5.2). This is done in addition to the normal processing on the Medisys side, where running averages are monitored for all alerts, etc.

Finally, UPHED identifies *events* as clusters of documents associated with labels, i.e. a set of diseases and locations describing clusters themselves, as already profusely shown in this chapter.

[6] http://medusa.jrc.it/m-eco?language=en

Results: Medisys has proven to be useful and effective for finding documents from a large number of Web sources [SFvdG+08]. After Medisys identifies documents where alerts fire, UPHED can deliver more overall information about the specific outbreak of the diseases reported in those documents. UPHED's task is to aggregate documents into larger units than alerts, namely *events*.

From Medisys we selected 29,850 articles. The selection was boosted gathering documents containing specific keywords and their synonyms, i.e. *EHEC, Malaria, Measles, Rubella*, and *Tuberculosis*. This selection was kindly provided by *JRC* (*Joint Research Centre, Ispra, Italy*). Articles were selected within the period under surveillance, i.e. May-June 2011.

From this specific batch taken from Medisys, we counted the number of documents overlapping with our collection which UPHED was applied to. We found that, out of 13,076 processed by our approach, 8,684 are the same articles.

On top of news articles, alerts were computed using the procedure previously described. In total 255 relevant alerts were found. For each alert, we considered the set of news articles associated and computed the overlaps with each of the 12 events (i.e. clusters) of articles extracted by UPHED and reported in Table 4.9 and Table 4.10. Also, a human expert judged the accuracy of the alert, by analyzing a sample of documents reported together to each alert, assigning it to one of the four following categories. For more details, we report the entire evaluation in Appendix 4.7.

The **first category** is identified in green in Appendix 4.7. The documents contained in the specific alert are the same news articles related to one particular event detected by UPHED. Within the alert, articles mentioned one disease outbreak event. Important to be noticed is that it is always the case where one alert, falling in such category, contains only a small subset of articles associated to one UPHED event. The reason is that Medisys computes alerts based on documents which exactly match the keywords pre-specified within the system, and it is not a semantic approach. Furthermore, the alerts categorized here contain documents simply treating just one or few diseases mostly only in the location mentioned by the pre-defined alert; thus, it is reasonable that these documents fall also in the event relevant to that particular location and disease. On the other hand, since UPHED detects events and, in a simplistic way, co-occurrences of feature-instances (i.e. keywords of several type), thus, each event can correspond to multiple alerts. As an example, let the reader observe in Table 4.11 as multiple alerts in the first category often are related to just one event; this is the case of all alerts considering the country Germany and several diseases related to event E_1. In conclusion, 20 alerts fell into this category and additional 11 alerts showed a big subset of documents related to one UPHED cluster.

The **second category** is identified in orange in Appendix 4.7. The Medisys alert was appropriate and was related to no events detected by UPHED. The reason why UPHED was not able to detect these events has to be retrieved in the explanation that there were few documents treating the particular outbreak. To let the reader better understand, we can consider the events reported in Table 4.9 and Table 4.10.

It is easy to identify that each event contains hundreds of document associated with, out of a total of less than 10, 000 articles of the entire dataset. Thus, it is difficult for the UPHED algorithm to detect events associated with only tens of documents. Of course, it is always important the proportion with respect to the entire set. On the other hand, Medisys was capable to trigger an alert since it detects a sudden increase in the number of articles for a given category and country, by comparing the statistics for the last day with the two-weeks rolling average. Thus, for Medisys it is sufficient a specific increasing of the standard deviation of the particular combination category-country under observation to generate an alert. In conclusion, 11 alerts fell into this category.

The **third category** is identified in yellow in Appendix 4.7. The third group consists of Medisys alerts reporting disease outbreaks happening in locations different from the one shown in the alert or just inappropriate. Articles falling in this category truly contain the disease name mentioned in the alert, but such disease is not related to the alert-country. Also, many of these documents contain the alert-country only to identify the location where the article was published or where the journalist had her newspaper-headquarter. Most of these articles described the situation in many other countries. As an example, let us consider in Table 4.11 the alert mentioning Canada and several diseases. In such a case, Canadian journalists described the outbreak of EHEC happening in the European countries. Thus, it is often the case where resources associated with alerts of this category are related to multiple UPHED's events. Furthermore, since these documents include many countries/locations, then they constitute elements of noise since one article can contribute in increasing the counter of different alerts. This is because alerts will match all the countries and diseases reported in them. In summary, this category contains cases where the disease name was mentioned but only to inform and report an outbreak burst somewhere else in the globe. In conclusion, 116 alerts fell into this category.

The **fourth category** is identified in light red in Appendix 4.7. The last group consists of articles not written in English-language or without overlap with UPHED's clusters. The reason has to be found in the fact that out of 29, 850 articles collected and processed by Medisys, we have an overlap with the dataset processed by UPHED of 8, 684 documents. Then, it is possible the case where one alert was generated mostly by articles not written in English, thus UPHED was not able to process most of them. Another possibility was that most of the articles in one alert were in English, but not contained in the overlap with UPHED's dataset. In conclusion, the rest of alerts fell into this category.

In addition and for sake of clarity, there are few cases in which the alert was assigned to two categories. In such a case, two colors highlighted the alert in Appendix 4.7. This happened, for instance, whenever a subset of the documents, contained in such alert, were totally related to one event detected (i.e. first category) by UPHED, while the remaining were related to other outbreaks not relevant for the alerts combination *country-disease* (third category). As an example, let us consider in Appendix

4.7 the alerts related to Spain. A subset of articles are referring correctly to events from UPHED, while other articles can be classified as third category since they are reporting the outbreak of the disease happening in other countries, but mentioning also the location Spain. Another example is when an alert was categorized within the second and the third category. In such a case, the alerts (e.g., *USA:WHO*) contained documents reporting the EHEC outbreak all over Europe (third category), while the rest were treating an outbreak of tuberculosis in USA not detected by UPHED (second category).

To give the reader an overview of the first three categories, Table 4.11 reports a sample of alerts providing their categorization and a short description. All the alerts reported in Figure 4.9 are presented in this table plus some additional alerts.

In conclusion, Medisys computes statistics based on exact matching of keywords in different languages. Medisys is not a semantic approach and is not able to detect alerts with several locations or diseases representing the same medical burst or outbreak. Also, since Medisys does not explicitly select for outbreaks, but for mentions of diseases in any context, it is expectable that many documents might cite diseases in contexts unrelated to epidemics and outbreaks.

On the contrary, UPHED semantically recognizes equal public health events and it clusters together documents treating the same topic. Furthermore, our approach computes medical events based on multiple diseases and locations at the same time, as can be observed in Table 4.9 and Table 4.10 by the labels extracted to describe each event.

The combination of the two initially independent systems, Medisys and UPHED, can lead to a stronger application offering users complementary functionalities. For disease outbreaks, which are covered by both systems, the combination can lead to additional advantages overtaking the drawbacks of both:

1. UPHED's computationally heavier method can only be applied to the document collection pre-filtered by Medisys.

2. The medical event extraction by UPHED can act as a filter for users to identify only disease outbreak reports.

3. UPHED is not a pattern matching based approach to extract events, thus it is not language dependant, as *PULS*-style [SFvdG+08] does developing event extraction grammars for many languages. UPHED can simply benefit from the categorization of news items by Medisys as useful tool for the analysis performed by our method. Actually, entities and their classes used by both systems, namely UPHED and Medisys, are overlapping.

These issues are to be tackled in future work. As a summarization, according to observations so far illustrated, we think that Medisys can be complemented with our UPHED to provide a better medical information system able to rely on a semantic detection approach.

Alerts	**Description**
Category 1: related to one cluster detected by UPHED	
Belgium:EscherichiacoliInfection	Related to event E_6, about documents treating the economical repercussion of the entire outbreak on the European market.
Belgium:CattleEFSA	Related to event E_9, about peak of measles cases in all Europe.
France:EscherichiacoliInfection, France:Coma	Related to E_4, about many cases of EHEC contagion in France with some case of coma.
Germany:Epidemic, Germany:EscherichiacoliInfection, Germany:Fever, Germany:Foodborne	Related to E_1, about EHEC cases in Germany.
India:Malaria, India:WHO	Related to E_{10}, about India's scientific community is all set to launch a research programme on how to combat vector-borne diseases, after many cases of malaria.
Spain:EscherichiacoliInfection	Related to E_2, about EHEC contagion which was caused by contaminated Spanish vegetables. Since the entire Europe was talking about Spains vegetable as the reason of EHEC outbreak, this alert contains also many other documents reporting EHEC cases in Europe, but the biggest overlap is with E_2.
Category 2: related to events not detected by UPHED	
China:AIDS-HIV	Cases of AIDS-HIV in China. Too few documents to be detected by UPHED.
Japan:CattleEFSA, Japan:Epidemic, Japan:EscherichiacoliInfection	EHEC cases of different nature compared to those occurred in Europe. The burst was caused by infected meat served in a restaurant chain. Few documents to be detected by UPHED.
UnitedKingdom:Enterohaemorrhagic, UnitedKingdom:Haemorrhage, UnitedKingdom:Tuberculosis	Cases of tuberculosis in UK.
Category 3: alerts reporting disease outbreaks happening in different locations.	
Austria:EscherichiacoliInfection, Austria:Foodpoisoning, Austria:WHO	Austria was always mentioned with other European countries for statistics on EHEC infection. In some case were reported Austrians visiting Germany and reporting EHEC.
Bulgaria:EscherichiacoliInfection	Bulgarian authorities specially worried about EHEC in Germany and reporting cases in the continent.
Canada:Epidemic, Canada:EscherichiacoliInfection, Canada:Hospital	In Canada, many news reporting the European EHEC outbreak. Also, Canada launched food inspection on food coming from EU.
Germany:CattleEFSA, Germany:Coma, Germany:Diarrhoea, Germany:Communicabledisease, Germany:Foodpoisoning	Report on EHEC cases all over Europe.

Table 4.11 Sample of alerts categorized together with a description. Mentioned events are related to clusters detected in Section 4.5.8. All alerts reported in Figure 4.9 are presented

4.6 Conclusions

In this research, we observed and exploited two main characteristics of text documents, i.e. their content and timestamps, to build up an approach for clustering articles in events with an unsupervised learner. Both the contents and the time information of articles are modeled explicitly and effectively. Especially the model of timestamps works like auto-adaptive sliding windows on time line, which overcomes the inflexible usages of timestamps in traditional retrospective event detection algorithms. Our method incorporates two main techniques: the burst function analysis and the entity-centric feature representation. Also, such a burst function analysis and entity-centric feature representation were combined in a generative model that is the basis of the algorithm. The model was refined for representing periodic, non-burst features with the Cauchy-Lorentz distribution. The evaluations showed that better sampling is reached by such distribution which resulted also in better efficiency of the algorithm. The algorithm is easy to implement in practice.

Furthermore, we prove the goodness of our theoretical study adapting our unsupervised learner to the medical domain, extracting a particular instance of events within the context of Epidemic Intelligence: the public health event PHE. More specifically, the adaptations included the consideration of domain specific features that allow to detect only domain-specific events.

In order to analyze the results of the introduced method, we ran several experiments. For the specific task considered, i.e. public health event detection, no annotated data set is available. Anyway, we performed some analysis on real-world data sets. Finally, we run our method demonstrating its effectiveness detecting a recent outbreak of enterohemorrhagic Escherichia coli (EHEC) occurred in northern Germany starting in May 2011.

For the task of detecting events, our approach achieved good quality results for a precision of 60% and a recall of 71% on manually annotated data. We showed that better results are achievable when using an unsupervised event detection algorithm that is specifically adapted to the medical domain, when compared with the domain independent event detection or event extraction approaches based on information extraction. Also, a qualitative assessment of the events also presented that detected clusters correspond to real-world health events, that have been listed on the public bulletin of international agencies.

The approach has been designed to address the limitations health organizations criticize in existing systems for detecting events. As a result, the presented algorithm allows to detect both, events that are rare or reoccurring. For the medical domain, this allows public health officials to rely upon alternative sources to corroborate information about public health events, which is important, since a diversity of information sources can offer an additional means of mitigating the impact of potential threats.

4.7 Appendix

In this section, we provide details about the evaluation on *Alerts* extracted from Medisys. For each alert we consider all the associated documents; then, for each *Event* detected by UPHED and reported in Table 4.9 and Table 4.10, we consider all the associated documents and we compute the overlap of documents between alert-document-set and event-document-set. In the following table, on the first column we report alerts. Each alert-row intersects the columns where events are presented. The intersection identifies the number of documents in common (i.e. overlaps) between the corresponding alert and the respective event. For each overlap, a human expert analyzed a sample of articles and judged the accuracy of the alert. Thus, each alert was assigned to one of the four categories presented in Section 4.5.9 and hereafter identified by four different colors.

In the following Table are reported the **Level** of each alert, i.e. *High* or *Medium*, the **Date** of the alert's burst, the **Category** it belongs to, and the overlaps between alert's documents with the documents associated to each event. To summarize, each category was identified by a color:

1. Category is identified in green if the Medisys alert is related to one cluster detected by UPHED.

2. Category is identified in orange if the Medisys alert is appropriated and related to NO one cluster detected by UPHED.

3. Category is identified in yellow if Medisys alert reports disease outbreaks happening in locations different from the one shown in the alert or just inappropriate. This category contains cases where the disease name was mentioned but only to inform and to report an outbreak burst somewhere else in the world.

4. Category is identified in light red if Medisys alert groups together articles not written in English-language or without overlap with UPHED events.

Alerts	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	Date	Level

Legenda

Related to one cluster detected by UPHED

Related to one cluster NOT detected by UPHED

Related to several outbreaks in several countries

No English documents or No overlaps with clusters by UPHED

Afghanistan:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	02.06.2011	High
AmericanSamoa:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	07.06.2011	Medium
Australia:disease	0	0	0	0	0	0	0	0	2	0	1	8	03.06.2011	High
Austria:Diarrhoea	1	0	0	1	0	0	0	0	0	0	0	0	02.06.2011	High
Austria:ECDC	0	0	0	0	0	0	0	0	0	0	0	0	02.06.2011	Medium
Austria:Enterohaemorrhagic	0	0	0	0	0	0	0	0	0	0	0	0	01.06.2011	Medium
Austria:Epidemic	0	0	0	0	0	0	0	0	0	0	0	0	02.06.2011	High
Austria:EscherichiacoliInfection	20	16	1	10	17	3	0	0	6	0	0	0	31.05.2011	High
Austria:Fever	0	0	0	0	0	0	0	0	0	0	0	0	04.06.2011	High
Austria:Foodpoisoning	13	13	0	5	12	0	0	0	7	0	0	0	02.06.2011	Medium
Austria:Hospital	0	0	0	0	0	0	0	0	0	0	0	0	31.05.2011	High
Austria:WHO	88	77	2	34	77	0	9	0	41	0	6	0	31.05.2011	High
Azerbaijan:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	07.06.2011	High
Belarus:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	21.06.2011	High
Belgium:CattleEFSA	0	0	0	0	0	0	0	0	3	0	0	0	16.06.2011	High
Belgium:Communicabledisease	1	1	0	0	0	1	0	0	2	0	0	0	07.06.2011	High
Belgium:Diarrhoea	0	0	0	0	0	0	0	0	0	0	0	0	25.06.2011	Medium
Belgium:ECDC	2	2	0	0	2	2	0	0	2	0	0	0	02.06.2011	Medium
Belgium:Enterohaemorrhagic	0	0	0	0	0	0	0	0	0	0	0	0	01.06.2011	Medium
Belgium:Epidemic	0	0	0	0	0	0	0	0	0	0	0	0	02.06.2011	High
Belgium:EscherichiacoliInfection	5	4	0	0	1	22	1	0	2	0	0	0	31.05.2011	High
Belgium:Food-borne	17	16	16	0	1	17	0	0	3	0	0	0	03.06.2011	Medium
Belgium:Foodpoisoning	5	5	0	0	0	5	0	0	2	0	0	0	16.06.2011	High
Belgium:Haemorrhage	0	0	0	0	0	0	0	0	0	0	0	0	01.06.2011	High
Belgium:Hospital	0	0	0	0	0	0	0	0	0	0	0	0	16.06.2011	High
Belgium:Pharmaceuticals	0	0	0	0	0	0	0	0	0	0	0	0	03.06.2011	High
Belgium:WHO	9	8	9	0	1	9	0	0	4	0	0	1	02.06.2011	High
Brazil:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	25.06.2011	High
Bulgaria:EscherichiacoliInfection	9	5	3	0	4	0	3	0	0	0	0	0	31.05.2011	Medium
Canada:Epidemic	1	1	1	0	0	0	0	0	1	0	0	0	07.06.2011	High
Canada:EscherichiacoliInfection	15	12	6	0	5	0	3	0	7	0	1	0	02.06.2011	High
Canada:Hospital	10	4	8	2	4	0	0	0	0	0	0	0	16.06.2011	Medium
Chile:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	01.06.2011	High
China:AIDS-HIV	0	0	0	0	0	0	0	0	0	1	10	0	03.06.2011	Medium
China:EscherichiacoliInfection	3	2	0	0	2	0	0	0	2	0	0	0	02.06.2011	High
China:Hospital	1	0	0	0	0	0	0	0	0	0	2	0	04.06.2011	Medium
China:Pharmaceuticals	4	1	0	1	2	0	0	0	2	0	3	2	03.06.2011	High
China:Tuberculosis	0	0	0	0	0	0	0	0	0	1	8	0	19.05.2011	High
Croatia:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	16.06.2011	Medium
CzechRepublic:Enterohaemorrhagic	0	0	0	0	0	0	0	0	0	0	0	0	31.05.2011	High
CzechRepublic:Epidemic	0	0	0	0	0	0	0	0	0	0	0	0	28.06.2011	Medium
CzechRepublic:EscherichiacoliInfection	2	1	0	0	1	0	0	0	2	0	0	0	28.06.2011	High
CzechRepublic:Hospital	0	0	0	0	0	0	0	0	0	0	0	0	16.06.2011	High
CzechRepublic:WHO	0	0	0	0	0	0	0	0	0	0	0	0	04.06.2011	High
Denmark:Diarrhoea	0	0	0	0	0	0	0	0	0	0	0	0	02.06.2011	High
Denmark:ECDC	0	0	0	0	0	0	0	0	0	0	0	0	02.06.2011	Medium
Denmark:Enterohaemorrhagic	0	0	0	0	0	0	0	0	0	0	0	0	01.06.2011	High
Denmark:Epidemic	0	0	0	0	0	0	0	0	0	0	0	0	02.06.2011	High
Denmark:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	28.06.2011	High
Denmark:Haemorrhage	0	0	0	0	0	0	0	0	0	0	0	0	01.06.2011	Medium

Alerts	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	Date	Level
Denmark:Hospital	0	0	0	0	0	0	0	0	0	0	0	0	31.05.2011	High
Denmark:IntensiveCareUnit	0	0	0	0	0	0	0	0	0	0	0	0	03.06.2011	Medium
Denmark:WHO	0	0	0	0	0	0	0	0	0	0	0	0	02.06.2011	High
Egypt:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	30.06.2011	High
Estonia:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	07.06.2011	High
Finland:Enterohaemorrhagic	0	0	0	0	0	0	0	0	0	0	0	0	10.06.2011	Medium
Finland:EscherichiacoliInfection	3	0	0	0	0	0	0	0	1	0	0	0	28.06.2011	Medium
Finland:WHO	0	0	0	0	0	0	0	0	0	0	0	0	04.06.2011	Medium
France:CattleEFSA	0	0	0	0	1	0	1	6	6	0	0	0	17.06.2011	Medium
France:Coma	2	0	0	8	0	0	2	0	0	0	0	0	18.06.2011	High
France:Diarrhoea	0	0	0	0	0	0	0	0	0	0	0	0	16.06.2011	High
France:ECDC	0	0	0	0	0	0	0	0	0	0	0	0	30.06.2011	High
France:Epidemic	0	0	0	0	0	0	0	0	0	0	0	0	16.06.2011	High
France:EscherichiacoliInfection	2	1	1	9	1	1	2	0	0	0	0	0	16.06.2011	High
France:WHO	0	0	0	0	0	0	2	1	2	1	0	0	02.06.2011	High
Georgia:EscherichiacoliInfection	1	1	0	0	0	0	0	0	1	0	0	0	08.06.2011	Medium
Germany:CattleEFSA	23	15	2	1	4	0	0	0	13	0	0	0	16.06.2011	High
Germany:Coma	23	12	0	4	12	1	1	0	7	0	0	0	17.06.2011	High
Germany:Communicabledisease	49	24	13	5	21	1	3	2	37	0	4	4	23.06.2011	High
Germany:Diarrhoea	75	47	14	5	39	9	0	0	22	0	0	0	02.06.2011	High
Germany:ECDC	28	23	1	0	19	2	0	0	7	0	0	0	31.05.2011	High
Germany:Enterohaemorrhagic	0	0	0	0	0	0	0	0	0	0	0	0	31.05.2011	High
Germany:Epidemic	13	1	5	0	3	0	0	0	3	0	0	0	31.05.2011	High
Germany:EscherichiacoliInfection	314	54	10	3	30	5	3	1	26	0	3	0	31.05.2011	High
Germany:Fever	45	0	10	0	11	0	0	0	9	0	0	0	31.05.2011	High
Germany:Food-borne	73	2	0	0	1	0	0	1	9	0	0	0	02.06.2011	High
Germany:Foodpoisoning	37	37	1	10	30	4	0	0	12	0	0	0	31.05.2011	High
Germany:Haemorrhage	0	0	0	0	0	0	0	0	0	0	0	0	31.05.2011	Medium
Germany:Hospital	52	8	7	3	15	0	0	1	21	0	1	0	16.06.2011	High
Germany:IntensiveCareUnit	0	0	0	0	0	0	0	0	0	0	0	0	31.05.2011	High
Germany:MRSA	0	0	0	0	0	0	0	0	0	0	0	0	03.06.2011	High
Germany:Pathogens	11	8	0	1	5	0	0	0	6	0	0	0	01.06.2011	Medium
Germany:Pharmaceuticals	7	4	0	1	4	0	0	0	6	0	2	2	03.06.2011	High
Germany:Travel	6	2	0	1	1	0	0	0	4	0	0	0	03.06.2011	Medium
Germany:TravelHealth	16	11	0	2	3	0	0	0	8	0	0	0	02.06.2011	High
Germany:UnknownDisease	0	0	0	0	0	0	0	0	0	0	0	0	04.06.2011	High
Germany:WHO	163	112	34	17	105	9	9	2	55	0	7	1	31.05.2011	High
Ghana:Malaria	0	0	0	0	0	0	0	0	0	0	1	0	21.06.2011	Medium
Greece:Epidemic	0	0	0	0	0	0	0	0	1	0	0	0	07.06.2011	High
Greece:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	03.06.2011	High
HongKong:EscherichiacoliInfection	4	1	0	0	0	0	0	0	0	0	0	0	03.06.2011	Medium
Hungary:Epidemic	0	0	0	0	0	0	0	0	0	0	0	0	01.06.2011	High
Hungary:WHO	8	0	8	0	8	0	0	0	1	0	0	0	04.06.2011	Medium
India:EscherichiacoliInfection	1	0	0	0	0	0	0	0	2	1	0	0	25.06.2011	Medium
India:Malaria	0	0	0	0	0	0	0	0	0	6	3	0	09.06.2011	High
India:WHO	0	0	0	0	0	0	0	0	0	3	2	0	02.06.2011	Medium
Ireland:EscherichiacoliInfection	14	9	9	0	3	9	2	0	6	0	2	2	25.06.2011	High
Israel:EscherichiacoliInfection	3	3	2	0	2	0	0	0	0	0	0	0	03.06.2011	Medium
Italy:Diarrhoea	12	2	10	0	12	0	0	0	2	0	0	0	31.05.2011	Medium
Italy:Epidemic	12	2	10	0	12	0	0	0	2	0	0	0	31.05.2011	Medium
Italy:EscherichiacoliInfection	22	12	10	0	20	8	9	0	2	0	0	0	31.05.2011	High
Italy:Fever	2	1	1	0	1	0	15	0	1	0	0	0	04.06.2011	High
Italy:Hospital	0	0	0	0	0	0	0	0	0	0	0	0	16.06.2011	High
Italy:WHO	2	2	1	0	1	0	13	0	2	0	0	0	02.06.2011	High
Japan:CattleEFSA	1	1	0	0	0	0	0	0	0	0	0	0	03.06.2011	High
Japan:Epidemic	1	1	0	1	1	0	0	0	0	0	0	0	07.06.2011	High
Japan:EscherichiacoliInfection	20	15	0	2	13	5	2	0	11	0	0	0	31.05.2011	High

Alerts	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	Date	Level
Kazakhstan:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	10.06.2011	Medium
Kenya:Malaria	0	0	0	0	0	0	0	0	0	0	0	0	17.06.2011	Medium
Latvia:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	03.06.2011	Medium
Lebanon:EscherichiacoliInfection	9	0	9	0	9	0	0	0	1	0	0	0	04.06.2011	Medium
Lebanon:WHO	8	0	8	0	8	0	0	0	0	0	0	0	04.06.2011	High
Libya:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	22.06.2011	Medium
Lithuania:EscherichiacoliInfection	7	7	7	7	0	0	0	0	0	0	0	0	28.06.2011	High
Luxemburg:ECDC	4	3	0	0	3	3	0	0	0	0	0	0	07.06.2011	Medium
Luxemburg:Epidemic	0	0	0	0	0	0	0	0	1	0	0	0	07.06.2011	High
Luxemburg:EscherichiacoliInfection	27	20	5	0	14	16	0	4	25	0	0	0	28.06.2011	High
Luxemburg:Hospital	0	0	0	0	0	0	0	0	0	0	0	0	07.06.2011	Medium
Luxemburg:WHO	3	0	1	0	0	3	0	0	4	0	0	0	03.06.2011	Medium
Mexico:EscherichiacoliInfection	5	5	0	0	5	0	0	0	5	0	0	0	01.06.2011	High
Mexico:WHO	5	5	0	0	5	0	0	0	5	2	2	0	04.06.2011	High
Netherlands:Diarrhoea	5	5	0	0	5	0	0	0	0	0	0	0	02.06.2011	High
Netherlands:Enterohaemorrhagic	0	0	0	0	0	0	0	0	0	0	0	0	31.05.2011	High
Netherlands:Epidemic	0	0	0	0	0	0	0	0	0	0	0	0	10.06.2011	Medium
Netherlands:EscherichiacoliInfection	26	26	0	6	24	0	0	0	7	0	0	0	31.05.2011	High
Netherlands:Foodpoisoning	9	8	0	5	7	0	0	0	3	0	0	0	16.06.2011	High
Netherlands:Haemorrhage	0	0	0	0	0	0	0	0	0	0	0	0	01.06.2011	High
Netherlands:Hospital	2	0	0	0	0	0	0	0	0	0	0	0	16.06.2011	High
Netherlands:WHO	5	5	0	0	5	0	0	0	0	0	0	0	31.05.2011	High
Norway:Diarrhoea	1	1	1	0	1	0	0	0	0	0	0	0	03.06.2011	Medium
Norway:ECDC	5	4	0	2	1	1	0	0	0	0	0	0	03.06.2011	Medium
Norway:Epidemic	2	2	0	2	2	0	0	0	0	0	0	0	03.06.2011	High
Norway:EscherichiacoliInfection	43	37	1	5	36	0	4	0	25	0	4	0	31.05.2011	High
Norway:Foodpoisoning	25	23	0	7	23	0	0	0	16	0	0	0	02.06.2011	High
Norway:Hospital	0	0	0	0	0	0	0	0	0	0	0	0	04.06.2011	High
Norway:Pathogens	15	12	0	8	12	0	0	0	6	0	0	0	04.06.2011	Medium
Norway:WHO	59	49	2	14	49	0	6	0	32	0	6	0	02.06.2011	High
Pakistan:EscherichiacoliInfection	0	0	0	0	0	0	0	0	1	0	0	0	08.06.2011	Medium
Poland:ECDC	3	2	0	2	1	1	0	0	0	0	0	0	07.06.2011	High
Poland:Epidemic	0	0	0	0	0	0	0	0	0	0	0	0	03.06.2011	Medium
Poland:EscherichiacoliInfection	13	8	4	8	2	2	2	2	3	0	2	0	28.06.2011	Medium
Poland:Foodpoisoning	0	0	0	0	0	0	0	0	0	0	0	0	02.06.2011	Medium
Portugal:EscherichiacoliInfection	1	1	1	0	1	0	0	0	0	0	0	0	31.05.2011	Medium
Romania:Epidemic	0	0	0	0	0	0	0	0	0	0	0	0	07.06.2011	High
Romania:EscherichiacoliInfection	4	2	4	0	0	1	0	0	3	0	0	0	07.06.2011	Medium
Russia:ECDC	0	0	0	0	0	0	0	0	12	0	0	0	02.06.2011	High
Russia:Enterohaemorrhagic	0	0	0	0	0	0	0	0	0	0	0	0	02.06.2011	Medium
Russia:Epidemic	5	1	5	0	2	0	0	0	20	0	0	0	02.06.2011	High
Russia:EscherichiacoliInfection	48	41	66	6	21	7	4	2	35	0	2	0	31.05.2011	High
Russia:Fever	0	0	10	0	8	0	0	0	3	0	0	0	04.06.2011	High
Russia:Food-borne	13	12	13	0	0	8	3	0	1	0	0	0	03.06.2011	High
Russia:Foodpoisoning	5	2	5	0	0	0	1	0	3	0	0	0	02.06.2011	High
Russia:Haemorrhage	0	0	0	0	0	0	0	0	0	0	0	0	02.06.2011	High
Russia:Hospital	0	0	0	0	0	0	0	0	0	0	0	0	01.06.2011	Medium
Russia:Pharmaceuticals	0	0	0	0	0	0	0	0	1	0	0	0	03.06.2011	High
Russia:WHO	29	22	29	0	7	9	8	0	15	0	4	1	02.06.2011	High
Samoa:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	07.06.2011	Medium
SaudiArabia:WHO	1	0	1	0	0	0	0	0	0	0	0	0	10.06.2011	Medium
Serbia:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	01.06.2011	High
Slovakia:Enterohaemorrhagic	0	0	0	0	0	0	0	0	0	0	0	0	01.06.2011	High
Slovakia:Epidemic	0	0	0	0	0	0	0	0	0	0	0	0	09.06.2011	High
Slovakia:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	31.05.2011	High
Slovakia:WHO	0	0	0	0	0	0	0	0	0	0	0	0	09.06.2011	Medium
Slovenia:Enterohaemorrhagic	0	0	0	0	0	0	0	0	0	0	0	0	01.06.2011	High

Alerts	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	Date	Level
Slovenia:Epidemic	0	0	0	0	0	0	0	0	0	0	0	0	08.06.2011	Medium
Slovenia:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	28.06.2011	Medium
SouthKorea:EscherichiacoliInfection	7	7	0	0	5	0	0	0	7	0	1	0	28.06.2011	Medium
Spain:CattleEFSA	1	4	1	0	3	0	0	0	5	0	0	0	02.06.2011	Medium
Spain:Communicabledisease	25	25	2	5	16	1	2	0	19	0	0	0	03.06.2011	High
Spain:Diarrhoea	11	11	0	1	9	0	0	0	4	0	0	0	02.06.2011	High
Spain:ECDC	22	22	0	0	19	2	0	0	5	0	0	0	31.05.2011	High
Spain:Enterohaemorrhagic	0	0	0	0	0	0	0	0	0	0	0	0	31.05.2011	High
Spain:Epidemic	1	1	1	0	0	0	0	0	0	0	0	0	31.05.2011	High
Spain:EscherichiacoliInfection	187	180	21	21	118	15	2	0	68	0	2	0	31.05.2011	High
Spain:Fever	20	6	10	0	16	0	0	0	7	0	0	0	31.05.2011	High
Spain:Food-borne	23	21	12	0	8	8	3	0	9	0	0	0	02.06.2011	High
Spain:Foodpoisoning	15	15	0	5	12	2	0	0	4	0	0	0	01.06.2011	High
Spain:Haemorrhage	0	0	0	0	0	0	0	0	0	0	0	0	31.05.2011	High
Spain:Hospital	2	2	0	0	0	0	0	0	2	0	0	0	31.05.2011	High
Spain:IntensiveCareUnit	0	0	0	0	0	0	0	0	0	0	0	0	31.05.2011	High
Spain:Pathogens	9	9	0	1	4	1	0	0	6	0	0	0	01.06.2011	Medium
Spain:Pharmaceuticals	6	6	0	1	2	0	0	0	6	0	0	0	02.06.2011	High
Spain:TravelHealth	10	8	0	2	3	0	0	0	3	0	0	0	02.06.2011	High
Spain:WHO	116	107	27	15	80	8	9	0	47	1	5	3	28.06.2011	High
Sweden:Antimicrobialresist	0	0	0	0	0	0	0	0	0	0	0	0	02.06.2011	High
Sweden:CattleEFSA	1	0	0	1	1	0	0	1	2	0	0	0	16.06.2011	High
Sweden:Communicabledisease	19	13	2	5	19	0	0	0	8	0	0	0	03.06.2011	Medium
Sweden:Diarrhoea	19	14	0	1	16	0	0	0	4	0	0	0	02.06.2011	High
Sweden:ECDC	15	14	0	0	14	2	0	0	3	0	0	0	31.05.2011	High
Sweden:Enterohaemorrhagic	0	0	0	0	0	0	0	0	0	0	0	0	31.05.2011	High
Sweden:Epidemic	6	3	2	1	6	0	0	0	2	0	0	0	31.05.2011	High
Sweden:EscherichiacoliInfection	170	129	8	21	145	18	2	1	10	0	3	0	31.05.2011	High
Sweden:Fever	30	6	10	0	23	0	0	0	11	0	0	0	03.06.2011	High
Sweden:Food-borne	22	18	9	1	11	9	0	0	8	0	0	0	03.06.2011	High
Sweden:Foodpoisoning	12	12	0	4	11	0	0	0	4	0	0	0	01.06.2011	High
Sweden:Haemorrhage	0	0	0	0	0	0	0	0	0	0	0	0	02.06.2011	High
Sweden:Hospital	14	1	4	2	13	0	0	1	10	0	1	0	16.06.2011	Medium
Sweden:Pharmaceuticals	0	0	0	0	0	0	0	0	0	0	0	0	03.06.2011	High
Sweden:TravelHealth	5	3	0	0	3	0	0	0	3	0	0	0	02.06.2011	Medium
Sweden:WHO	122	84	16	16	106	5	5	2	42	0	6	0	28.06.2011	Medium
Switzerland:Communicabledisease	2	2	0	2	2	0	0	0	0	0	0	0	03.06.2011	High
Switzerland:Diarrhoea	0	0	0	0	0	0	0	0	0	0	0	0	02.06.2011	High
Switzerland:ECDC	0	0	0	0	0	0	0	0	0	0	0	0	02.06.2011	High
Switzerland:Epidemic	2	2	0	2	2	0	0	0	0	0	0	0	02.06.2011	High
Switzerland:EscherichiacoliInfection	60	54	3	24	52	6	2	0	15	0	2	0	31.05.2011	High
Switzerland:Fever	0	0	0	0	0	0	0	0	0	0	0	0	04.06.2011	High
Switzerland:Haemorrhage	0	0	0	0	0	0	0	0	0	0	0	0	01.06.2011	High
Switzerland:Hospital	0	0	0	0	0	0	0	0	0	0	0	0	31.05.2011	High
Switzerland:Malaria	0	0	0	0	0	0	0	0	0	1	0	0	31.05.2011	Medium
Switzerland:WHO	88	77	2	34	77	0	6	0	37	4	6	0	02.06.2011	High
Syria:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	04.06.2011	Medium
Thailand:Epidemic	0	0	0	0	0	0	0	0	0	0	0	0	11.06.2011	Medium
Thailand:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	11.06.2011	High
Turkey:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	31.05.2011	High
Turkey:WHO	0	0	0	0	0	0	0	0	0	0	0	0	03.06.2011	High
Ukraine:EscherichiacoliInfection	0	0	0	0	0	0	0	0	0	0	0	0	02.06.2011	High
UnitedKingdom:Antimicrobialresist	0	0	0	0	0	0	0	0	2	0	2	2	03.06.2011	Medium
UnitedKingdom:CattleEFSA	5	5	0	0	2	0	0	0	4	0	0	0	22.06.2011	Medium
UnitedKingdom:Communicabledisease	15	9	6	1	5	0	0	1	19	0	4	4	03.06.2011	High
UnitedKingdom:Diarrhoea	17	16	0	0	12	0	0	0	5	0	0	0	02.06.2011	High
UnitedKingdom:ECDC	13	12	0	0	10	0	0	0	3	0	0	0	31.05.2011	High

Alerts	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	Date	Level
UnitedKingdom:Enterohaemorrhagic	0	0	0	0	0	0	0	0	0	0	0	0	31.05.2011	Medium
UnitedKingdom:Epidemic	1	1	1	0	0	0	0	0	0	0	0	0	31.05.2011	High
UnitedKingdom:EscherichiacoliInfection	65	40	2	1	44	2	2	1	12	0	3	0	25.06.2011	High
UnitedKingdom:Fever	4	0	2	0	0	0	0	0	2	0	0	0	03.06.2011	Medium
UnitedKingdom:Food-borne	5	2	0	1	3	1	0	1	2	0	0	0	07.06.2011	Medium
UnitedKingdom:Foodpoisoning	16	16	1	0	11	4	0	0	12	0	0	0	02.06.2011	Medium
UnitedKingdom:Haemorrhage	0	0	0	0	0	0	0	0	0	0	0	0	02.06.2011	High
UnitedKingdom:Hospital	4	2	0	0	2	0	0	0	4	0	0	0	31.05.2011	Medium
UnitedKingdom:Malaria	0	0	0	0	0	0	0	3	8	0	0	1	29.06.2011	Medium
UnitedKingdom:Pathogens	7	7	1	0	6	0	0	0	7	0	0	0	02.06.2011	High
UnitedKingdom:Pharmaceuticals	1	1	0	0	0	0	0	0	3	0	2	2	03.06.2011	High
UnitedKingdom:TravelHealth	4	2	0	0	1	0	0	0	1	0	0	0	03.06.2011	High
UnitedKingdom:tropicalmedicine	22	21	2	0	16	0	0	0	19	0	0	0	02.06.2011	Medium
UnitedKingdom:Tuberculosis	0	0	0	0	0	0	0	0	2	1	0	1	29.06.2011	High
UnitedKingdom:WHO	42	33	2	1	35	0	2	1	22	0	3	0	28.06.2011	High
USA:AIDS-HIV	0	0	0	0	0	0	0	0	0	0	2	0	09.06.2011	High
USA:CattleEFSA	9	10	2	0	0	0	0	0	6	0	0	0	03.06.2011	Medium
USA:Coma	9	5	0	2	5	0	1	0	5	0	0	0	17.06.2011	Medium
USA:Communicabledisease	13	10	2	5	6	1	0	1	7	0	2	0	03.06.2011	High
USA:Diarrhoea	34	12	11	6	19	0	0	1	1	0	1	0	03.06.2011	High
USA:ECDC	0	0	0	0	0	0	0	0	0	0	0	0	03.06.2011	Medium
USA:Epidemic	7	3	4	0	3	0	0	0	1	0	0	0	02.06.2011	High
USA:EscherichiacoliInfection	172	114	31	36	87	16	2	3	2	0	3	1	28.06.2011	Medium
USA:Fever	21	4	10	2	14	0	0	2	2	0	0	0	03.06.2011	High
USA:Food-borne	2	1	0	0	0	0	0	1	7	0	0	0	07.06.2011	High
USA:Foodpoisoning	10	10	1	0	7	2	0	0	8	0	0	0	02.06.2011	Medium
USA:Hospital	0	0	0	0	0	0	0	0	0	0	0	0	25.06.2011	Medium
USA:Malaria	0	0	0	0	0	0	0	0	0	2	2	10	26.05.2011	High
USA:Pathogens	24	19	2	7	15	0	0	0	2	0	1	0	02.06.2011	High
USA:Pharmaceuticals	2	0	0	0	0	0	0	0	3	0	0	0	23.06.2011	High
USA:Salmonellosis	2	0	0	0	0	0	0	0	5	0	0	0	08.06.2011	Medium
USA:Tuberculosis	0	0	0	0	0	0	0	0	12	1	4	0	22.06.2011	Medium
USA:WHO	111	86	26	30	79	0	0	0	55	11	13	1	28.06.2011	Medium
Yemen:EscherichiacoliInfection	1	1	0	0	1	0	0	0	0	0	0	0	04.06.2011	High

5 Conclusions and Outlook

In this thesis, we presented various approaches on Clustering Information Entities based on Statistical Methods, with the goal of providing useful advices and references to fundamental concepts accessible to the broad community of clustering practitioners.

We described three important applications of clustering algorithms in Information Retrieval: (1) *Similarity Search for High Dimensional Data Points*, with the purpose to find Near Duplicate Images; (2) *Measuring Latent Variables in Social Sciences*, with the aim to visualize Research Communities; and (3) *Generative Model for Content Analysis of Natural Language Documents* to detect Events. Also, we introduced three different cluster definitions according to the clustering algorithm's purpose and domain where they applied to.

In this section, we first summarize our major contributions with respect to the three above mentioned domains, and then we discuss some issues which remain open for future investigations.

5.1 Summary of Contributions

Similarity Search for High Dimensional Data Points. In chapter 2, we considered the problem of detecting near-duplicates for high dimensional data points in an incremental manner, which is a domain focused application of the more general problem of answering range search queries.

The presented application scenario of this work was online near duplicate detection for multimedia content sharing websites such as Flickr [Fli] and Youtube [You]. In detail, whenever a user is uploading an image or a video, it would be desirable if near-duplicates that are very similar (content-wise) to the one being uploaded can be retrieved and returned to the user in real-time. In this way, the user can identify redundant copies of the object promptly and decide if she should continue the upload.

For our work, we leveraged a well known technique, Locality Sensitive Hashing (LSH), and we proposed a new approach easy to implement and that preserves the LSH theoretical guarantee on the quality of the search result.

Also, we concentrated on in-memory index structure since fast real-time response is the first priority in the applications we considered. For high dimensional similarity search, the index size can be as large as, or even larger, than the data set size in order to give an efficient query response time. Therefore, reducing the memory cost while providing fast response was the main concern of our approach.

Thorough experiments conducted on 3 real-world data sets demonstrated that our method consistently outperforms LSH in terms of query time in all cases we tried, with a small amount of extra memory cost. To achieve the same query time saving, we showed that LSH needed significantly more space.

Measuring Latent Variables in Social Sciences. In chapter 3, we studied the problem of visualizing technology-enhanced learning (TEL) research communities, and we addressed the issue of detecting communities and sub-communities in the area.

The motivation was that many different conferences and journals are devoted to different aspects of technology-enhanced learning, providing a variety of forums through which publish research results. The downside of this variety is, however, that TEL is a much more fragmented area than most of other research areas, making it difficult to gain an overview of recent advances in the field, even for experienced researchers.

With our approach, we boosted in the direction of increasing synergies between different sub-areas and researchers, and, last but not least, of providing funding agencies with evidence of new research results, innovative applications, and promising new approaches for TEL. Also, we provided a first step towards this goal, by employing the technique of Author Co–citation Analysis (ACA) on the large subset of TEL conferences related to computer science as indexed by DBLP [DBLa] and CiteseerX [Cit] – the latter provides citation information for each indexed paper. In fact, we leveraged ACA which relies on the insight that, if two authors are cited together very often in scientific articles, their work must be related to the same research field. In the last analysis, we described our methodology for data collection, solutions for problems that we encountered, and the techniques of author co-citation and factor analysis for detecting communities in a given research area.

The results are promising and show the method's potential as regards mapping and visualizing TEL research communities, making researchers aware of the different research communities relevant for technology-enhanced learning, and thus better able to bridge communities wherever needed.

Generative Model for Content Analysis of Natural Language Documents. In chapter 4, we considered the problem of clustering articles in an unsupervised manner. Unsupervised learning means no supervision, thus there is no human expert who assigned documents to classes. Our purpose to cluster documents was to extract events, where an event was defined as a specific thing happening at a specific time and place, which may be consecutively reported by many articles in a period under observation. Our method is part of the Retrospective Event Detection area (RED). RED is defined as the discovery of previously happened events in an historical corpus.

The presented application scenario of our methodology was the specific and important domain of health which boosted us to apply our approach to real needs in the medical area. In particular, the detected events we extracted are defined as Public Health Event (PHE). Actually, a PHE is intended to be some emerging infection, symptom, or illness affecting people or animals in a particular geographic place during a specific time period.

Our work leveraged on our intuition that articles contain two kinds of information: contents and timestamps. The usefulness of time information is often ignored, or at least time information is used in unsatisfied manners. According to these observations, we explored RED and considered the better representations of news articles and events, which effectively modeled both the contents and the time information. Especially the model of timestamps worked like auto-adaptive sliding windows on time line, which overcame the inflexible usages of timestamps in traditional retrospective event detection algorithms. Our method incorporated two main techniques: the burst function analysis and the entity-centric feature representation. Also, such a burst function analysis and entity-centric feature representation were combined in a generative model that was the basis of the algorithm. The model was refined for representing periodic, non-burst features with the Cauchy-Lorentz distribution. The evaluations showed that better sampling is reached by such distribution which resulted also in better efficiency of the algorithm.

In conclusion, we proved the goodness of our theoretical study. In order to analyze the results of the introduced method, we ran several experiments. For the specific task considered, i.e. public health event detection, no annotated data set is available. Anyway, we performed several analyses on real-world data sets. Finally, we run our method demonstrating its effectiveness detecting a recent outbreak of enterohemorrhagic Escherichia coli (EHEC) occurred in northern Germany starting in May 2011.

5.2 Open Directions

Similarity Search for High Dimensional Data Points. While much progress in Content Based Image Retrieval technology has been realized in recent years, there are still many open research issues. One important issue is how to achieve effective high-dimensional indexing to support search in large image collections. Considering our contribution, in-memory indexing techniques are ideal solutions to speed up the searching process if the help of disk-based index are not necessary since a disk access is an order of magnitude slower than a memory operation. As presented in our work, for a data set with 1 million points, an index storing all the point IDs once only needs 12MB memory assuming that each ID takes 12 bytes; if each point is a 162-dimensional point and each dimension of a point takes 4 bytes, storing all the points needs 648MB, which is tolerable even for an inexpensive PC nowadays. An open direction we will investigate is how to process Web-scale data sets with billions of points; this process might need clusters with tens or hundreds of distributed machines, and the task for the future will be to make our approach scalable and distributed on several servers.

In conclusion, as the World Wide Web continues to expand, web-based image search engines become increasingly desirable. Although there are a number of very successful text-based search engines, such as Yahoo, Google, Alta Vista, and so forth, web-based image search engines are still in their infancy. More technical breakthroughs are required to make image search engines as successful as their text-based counterparts.

Measuring Latent Variables in Social Sciences. Any fair evaluation of citation analysis, as an aid in assessing scientists, must acknowledge that there is much about meaning of citation rates than we know [Gar79]. We are still imprecise about the quality of the scientific performance they measure. We still know very little about how sociological factors affect citation rates. There is still much uncertainty about all possible reasons for low citation rates. And there is still much to learn about the variations in citation patterns from field to field.

On the other hand, we know that citation rates say something about the contributions made by an individual's work, at least in terms of the utility and the interest the rest of the scientific community finds in it.

As an important next step, we will try to delineate the importance of citations rates, including our outcomes for future research in ACA. Also, we will extend our dataset with additional publication and citation data relevant for TEL, most importantly education and psychology, as relevant for example for computer supported collaborative learning (e.g., the CSCL conference is not indexed in DBLP and therefore missing in our analysis).

Finally, ACA studies will take advantage of emerging data sources and tools and we will combine recent advanced information visualization techniques with various co-citation counting methods to produce even more interesting and revealing ACA results.

Generative Model for Content Analysis of Natural Language Documents. The instance-based nature of the presented approach needs to be considered as a limitation of the work. Since no model is built that can be reused on new data, the event detection needs to be re-run for each new set of data. In future work, we plan to consider an on-line alternative to this instance-based approach to allow for a continuous event detection, for example, using Web data streams as sources.

In order to simplify our model, we assumed that all kinds of information of the i-th article, given an event e_j, are conditional independent. As a future step, this statement will be corroborated or confuted with further analysis. Of particular interest for the medical scenario will be discovering if *disease* and *time* are conditional independent or not. Exploiting the achieved results, we will update our approach. Also, we will try to infer if the conditional independence of all kinds of information contained within articles depends by the application domain.

Interesting for the future will be to discover trends in online web search query data. Google Correlate [Cor] is an automatic method for query/term selection, useful for our approach to estimate the true value of an event/phenomenon.

Concentrating into our motivating example, given the exploratory nature of EI, better mechanisms are needed to support the aggregation of events for statistical models, as well as the navigation for epidemic intelligence gathering. On the one hand, public health officials are only interested in receiving a limited number of events per session; yet on the other hand, they want to be adequately informed, and not miss potentially relevant ones. Identifying the balance for this trade-off is a challenge which is currently not handled by any of the existing tools.

Finally, open for future work is a more detailed and robust evaluation which requires an annotated corpus as a benchmark and would allow to run evaluations with varying feature sets and settings. We plan to develop such a data set with the input by domain experts, to support a more robust evaluation of our approach, but also of other systems in this domain and to allow comparison of various approaches.

Curriculum Vitae

Marco Fisichella

Birth date and city: 10/09/1982 Reggio Calabria (Italy)

Mobile: (+49) 178 6252380
Office: (+49) 511 762 17710
Email: fisichella@L3S.de

WORK EXPERIENCE

Invited Reviewer and PC member at:

1. The 2-nd **International Conference on Emerging Intelligent Data and Web Technologies**, 2011, Tirana (Albany) - on the track area about Knowledge Discovery and Data Mining
2. **Data & Knowledge Engineering Journal** - Elsevier - on the track area about Reasoning Approaches
3. **Information Sciences Journal** - Elsevier
4. The 12th **International Conference on Mobile Data Management**, **MDM** 2011, Luleå (Sweden) - on the track area about Pervasive Data Management
5. The 1-st **international workshop on linked web data management**, **LWDM** 2011, in conjunction with **EDBT** 2011, Uppsala (Sweden)
6. The 2011 IEEE / WIC / ACM International Conferences on **WEB INTELLIGENCE**, **WI** 2011, Lyon (France)
7. The 2011 **IEEE International Conference on Data Mining series**, ICDM, Vancouver (Canada)
8. The 27th **SIGAPP Symposium On Applied Computing, SAC** 2012, Riva del Garda (Italy)
9. The 17th **International Conference on Database Systems for Advanced Applications**, **DASFAA** 2012, Busan, South Korea
10. The **World Wide Web Conference**, WWW 2012, Lion, France

In compliance with Italian Law 196/2203, further amendments and integrations, I hereby consent to any legal use of all of the above personal data.

2011

L3S Research Center of Leibniz University of Hannover (June 2007, Present)
Description of my duties:

- **Project manager** in the **OpenScout** project (April 2010, Present) and **work package coordinator**. **Responsible** for the monetary resources disposed by the EU commission **(~400k euro** in 3 years) and **Coordinator** of 10 Partners.
  - OpenScout joins together 10 partners all around Europe. It is an EU project under the 7th framework. At L3S we lead the workpackage about the infrastructure, coordinating all partners involved. The project aims at accelerating the use, improvement and distribution of open content in the field of management education and training with a focus on SMEs and continuous training by providing skill-based search of content to large communities for learning – either in professional user communities (via integration with learning management systems) or to open web 2.0 communities (via integration to social network platforms).

- **Responsible** for the **WebRatio** Competence Center in Hannover (June 2007, December 2011). The WebRatio Competence Center offered a comprehensive service for conception, development and employment of complex data-driven system. In the context of the Competence Center, we carried out projects both in the context of internal IT applications, together with your IT department, and in the context of developments for companies' customers, together to centers with software developers and Web designers. Further we offered training courses and seminars.

- **Responsible** for realization, implementation, management of the front-end and back-end of the official web site http://www.L3S.de (June 2007 – June 2010).

- **Task manager** and **co-developer** of "Reference Reconciliation on Visual Objects" module in the **Pharos** project (June 2007 – January 2010).
  - Pharos project was within the 6th framework of the specific research and technological development program "Integrating and Strengthening the European Research Area (2006-2008)"; it is the acronym of Platform for Search of Audio-Visual Resources across Online Spaces.

- **Java programmer** in the **Rewerse** project (June 2007 – December 2007).
  - Rewerse project was within the 6th framework of the specific research and technological development program "Integrating and Strengthening the European Research Area (2006-2008)"; it is the acronym of REasoning on the WEb with Rules and Semantics.

Hannover (Germany)

CEBIT (March 2011)
Cebit is the foremost tradeshow for the digital industry.
Presentation at the expo of **M-eco project** - Medical System to detect health events from social media data and user generated content. Booth 17, Hall 9.

Hannover (Germany)

European Patent application EP 11425276.0 (November 2011)
Patent pending for "**SMART-FIRE**: an innovative Fire Alarm System"

Writer of the **book** "**Goodbye Mamma**", a guide for Italians to find opportunities abroad with a collection of experiences, tips, and sources.
More info at http://www.goodbyemamma.com/

2010

CEBIT (March 2010)
Presentation at the expo of **Pharos project** – Platform for search of audiovisual resources across online spaces. Booth 17, Hall 9.

Hannover (Germany)

Italian Patent application IT RC 2010A000001 (January 2010)
Patent pending for "**SMART-FIRE**: an innovative Fire Alarm System"

On **January 2011,** we won the **Price as Best Patent Request** in 2010; the concourse was called by Region Calabria - Italy On **October 2011,** we won the call about **Support to the processes of technological innovation on the part of Micro, Small and Medium Enterprises** in the province of Reggio Calabria - Edition 2010.	
	2009
Expo Matching (November 2009) Invitation to present my project "**SMART-FIRE**: an innovative Fire Alarm System" during the Fifth Edition of Matching called "Innovating, Going international", held at Fieramilano in Rho (MI) http://www.matchingtuttolanno.it/cdo/areaPubblica/home.seam.	**Milano (Italy)**
Start Cup Milano Lombardia 2009 Participation to the challenge for *business plan competition* promoted by *six universities of the Lombardy region* with the support of the strategic partner *Vodafone Italia* to encourage and support the *creation of high-tech start-ups*. With my team and our project on "**SMART-FIRE**: an innovative Fire Alarm System" we reached the 12th places out of 200 participants.	**Milano (Italy)**
	Till 2008
IBM (February 2005 – December 2006) **Junior Consultant** – Development of the interdisciplinary complex innovation project about "Goods tracking and Risk Management" entailed by Alta Scuola Politecnica (ASP) for IBM Italy.	**Milano (Italy)**
Coware (November 2006 – December 2006) **Freelance Collaborator** Coware is the leading supplier of system-level electronic design automation (EDA) software and services. Coware is headquartered in San Jose, Calif. Description of my duties: – Analysis, assessment and engineering of the web customer support service. Realization and implementation of web customer support service.	**Aachen (Germany)**
ATM Azienda Trasporti (April 2004 – July 2004) **Junior Consultant** – Analysis, assessment and realization of the wireless network.	**Catania (Italy)**
Selkron (October 2001 – October 2004) **Junior Consultant** – Realization and implementation of the front-end and back-end of the web-site of a plant safety company Selkron in Reggio Calabria (www.selkron.it).	**Reggio Calabria (Italy)**

EDUCATION

Summer School on Multimedia Semantics (September 2008) The subject of active research during the school was the integration of knowledge, semantics and low-level multimedia processing for the purpose of automatic semantics extraction from multimedia content.	**Chania (Greece)**
Computer Engineering – Leibniz University of Hannover (June 2007, Present) **Ph.D. student** in "Clustering Information Entities based on Statistical Methods" under the guide of Prof. Dr. techn. Wolfgang Nejdl.	**Hannover (Germany)**
Computer Engineering - Politecnico di Milano (October 2004 - April 2007) **Master of Science in Computer Engineering.** Final mark 105 /110 Thesis title: "Supporting Flexible Processes in Project-Centered Learning: The ASP Platform"	**Milano (Italy)**

In compliance with Italian Law 196/2203, further amendments and integrations, I hereby consent to any legal use of all of the above personal data.

Supervisor Prof. Stefano Ceri.	
Computer Engineering - Politecnico di Torino (January 2005 - June 2007) **Master of Science in Computer Engineering.** Final mark 105 /110 – Thesis title: "Risk Management and Goods Tracking" - Supervisor Prof. Stefano Ceri, Prof. Verganti.	**Torino (Italy)**
Alta Scuola Politecnica - Politecnico di Milano & Torino (January 2005 - June 2007) **Courses** followed: (1) Innovation and society; (2) Actors and decisions in projects; (3) Problem setting in complex systems. Three **papers** published in the following fields: 1. Context and determinants of innovation. 2. Organizational structures and collaborations for the development of innovation. 3. The logic of modeling. The power of chaos. 4. Management of innovation. The value of the patent. **Alta Scuola Politecnica** is the school founded by Politecnico di Torino and Politecnico di Milano focusing on talent, innovation, interdisciplinary capabilities. It is attended by the top 2.5% of students enrolled in a master degree of the two universities. It is recognized by the Italian Minister of the Education (MIUR). This community of students coming from different disciplines are joined into one single class to develop their interdisciplinary capabilities to conceive and implement complex innovation projects. The ASP consists of an additional program that runs in parallel to the master of sciences. It entails interdisciplinary courses on innovation and design processes and the development of projects, in multidisciplinary teams, for external firms and institutions (Goods Tracking and Risk Management). For further information: www.asp-poli.it.	**Milano-Torino (Italy)**
Computer Engineering – University of Catania (October 2001 - July 2004) **Bachelor degree of Science in Computer Engineering.** Final mark 110 /110 Thesis title: "Development of a support system to the design and test of IEEE 802.11B Wireless Networks" - Supervisor Prof. Cavalieri.	**Catania (Italy)**
Liceo Scientifico Leonardo da Vinci (October 1996 - July 2001) **Diploma in Scientific Studies.** Final mark: 100 / 100	**Reggio Calabria (Italy)**

LANGUAGES

Italian (native)
English (fluent)
German (educational B1)
French (educational A1)

COMPUTER SKILLS

Operating systems
Programming (C, Java, Visual Basic, WebML, SQL, OQL, XQUERY)
Implementation of Web services using C and Java
Database Management (MySQL Server, Access)
Database modelling and integration
Image processing
Digital graphics
Office
Creation of Web sites (HTML, ASP, JSP, WebRatio tool)

INTERESTS

Since I'm working on EU projects and I started my Ph.D. at L3S, I started travelling a lot all around Europe and I love it. I like reading books and thinking about the hints and concepts and thoughts expressed within. I like painting, but I like more playing guitar with friends. I love open my mind and think with the others about everything. I think the diversity in the world is the salt of life, optimism is a way to face with life.

In compliance with Italian Law 196/2203, further amendments and integrations, I hereby consent to any legal use of all of the above personal data.

Bibliography

[AI05] Alexandr Andoni and Piotr Indyk. E^{2}LSH0.1 User Manual. http://web.mit.edu/andoni/www/LSH/manual.pdf, 2005.

[AI08] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM (CACM)*, 51(1), 2008.

[AIP06] Alexandr Andoni, Piotr Indyk, and Mihai Patrascu. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Annual Symposium on Foundations of Computer Science (FOCS)*, pages 459–468, 2006.

[All02] James Allan. Introduction to topic detection and tracking. pages 1–16, 2002.

[APL98] James Allan, Ron Papka, and Victor Lavrenko. On-line new event detection and tracking. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 37–45, New York, NY, USA, 1998. ACM.

[Arc] Telearn Archive. http://telearn.noe-kaleidoscope.org/.

[AWB03] James Allan, Courtney Wade, and Alvaro Bolivar. Retrieval and novelty detection at the sentence level. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 314–321, New York, NY, USA, 2003. ACM.

[BBJ$^+$00] Stefan Berchtold, Christian Böhm, H. V. Jagadish, Hans-Peter Kriegel, and Jörg Sander. Independent quantization: An index compression technique for high-dimensional data spaces. In *Proceedings of the 16th International Conference on Data Engineering (ICDE)*, pages 577–588, 2000.

[BC] IEEE TLT Board and Steering Committee. http://www.computer.org/portal/web/tlt/edboard.

[BCF03] Thorsten Brants, Francine Chen, and Ayman Farahat. A System for new event detection. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 330–337, New York, NY, USA, 2003. ACM.

[BCG05] Mayank Bawa, Tyson Condie, and Prasanna Ganesan. Lsh forest: self-tuning indexes for similarity search. In *Proceedings of the International Conference on World Wide Web(WWW)*, pages 651–660, 2005.

[Ben75] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM (CACM)*, 18(9), 1975.

[BKL06] Alina Beygelzimer, Sham Kakade, and John Langford. Cover trees for nearest neighbor. In *Machine Learning, Proceedings of the 23rd International Conference (ICML)*, pages 97–104, 2006.

[CC99] Chaomei Chen and Les Carr. Trailblazing the literature of hypertext: Author co-citation analysis. In *In Proceedings of the 10th ACM Conference on Hypertext and hypermedia*, pages 51–60. ACM Press, 1999.

[CCFS11] Alfredo Cuzzocrea, Juri Luca De Coi, Marco Fisichella, and Dimitrios Skoutas. Graph-based matching of composite owl-s services. In *DASFAA Workshops*, pages 28–39, 2011.

[CDK+08] Nigel Collier, Son Doan, Ai Kawazeo, Reiko Matsuda Goodwin, Mike Conway, Yoshio Tateno, Quoc Hung Ngo, Dinh Dien, Asanee Kawtrakul, Koichi Takeuchi, Mika Shigematsu, and Kiyosu Taniguchi. Biocaster: detecting public health rumors with a web-based text mining system, 2008.

[CF11a] Alfredo Cuzzocrea and Marco Fisichella. Discovering semantic web services via advanced graph-based matching. In *SMC*, pages 608–615, 2011.

[CF11b] Alfredo Cuzzocrea and Marco Fisichella. A flexible graph-based approach for matching composite semantic web services. In *EDBT/ICDT Workshop on Linked Web Data Management*, pages 30–31, 2011.

[CFM10] Juri Luca De Coi, Marco Fisichella, and Maristella Matera. Managing adaptivity in web collaborative processes using policies and user profiles. In *ICWE Workshops*, pages 150–162, 2010.

[Cit] CiteseerX. http://citeseerx.ist.psu.edu/.

[CNAM96] Farrington C.P., Andrews N.J., Beale A.D., and Catchpole M.A. A statistical algorithm for the early detection of outbreaks of infectious disease. *Journal of the Royal Statistical Society*, A:574–563, 1996.

[Cor] Google Correlate. http://www.google.com/trends/correlate/.

[CPIZ07] Ondrej Chum, James Philbin, Michael Isard, and Andrew Zisserman. Scalable near identical image and shot detection. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval (CIVR)*, pages 549–556, 2007.

[DAFK$^+$11] Ernesto Diaz-Aviles, Marco Fisichella, Ricardo Kawase, Wolfgang Nejdl, and Avaré Stewart. Unsupervised auto-tagging for learning object enrichment (best paper award). In *EC-TEL*, pages 83–96, 2011.

[DBLa] DBLP. http://www.informatik.uni-trier.de/~ley/db/.

[DBLb] DBLPVis. http://dblpvis.uni-trier.de/help/overview.html.

[DDAD$^+$11] Kerstin Denecke, Ernesto Diaz-Aviles, Peter Dolog, Tim Eckmanns, Marco Fisichella, Ricardo Gomez-Lage, Jens Linge, Pavel Smrz, and Avaré Stewart. The medical ecosystem [m-eco] project: Personalized event-based surveillance. In *IMED: International Meeting on Emerging Diseases and Surveillance*, 2011.

[DIIM04] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Symposium on Computational Geometry (SCG)*, pages 253–262, 2004.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society*, 39(1):1–38, 1977.

[Dp] Topic Detection and Tracking (TDT) project. http://www.nist.gov/speech/tests/tdt.

[DR06] Fan Deng and Davood Rafiei. Approximately detecting duplicates for streaming data using stable bloom filters. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 25–36, 2006.

[Dub93] Richard Dubes. *Cluster analysis and related issues.* In Handbook of Pattern Recognition & Computer Vision, C. H. Chen, L. F. Pau, and P. S. P. Wang, Eds. World Scientific Publishing Co., Inc., River Edge, NJ, 1993.

[ea09] D. Hartley et al. The landscape of international event-based biosurveillance. *Emerging Health Threats*, 2009.

[ECT] ECTEL. http://ariadne.cs.kuleuven.be/ectel/rankings.html.

[EM] ED-MEDIA. http://ariadne.cs.kuleuven.be/edmedia/rankings.html.

[FC12] Marco Fisichella and Alfedo Cuzzocrea. Improving flexibility of workflow management systems via a policy-enhanced collaborative framework. In *WEBIST: Proceedings of the International Conference on Web Information Systems and Technologies*, 2012.

[FDN] Marco Fisichella, Fan Deng, and Wolfgang Nejdl. Similarity search for high dimensional data points. In *TKDE: Under Submission at IEEE Transactions on Knowledge and Data Engineering Journal*.

[FDN10] Marco Fisichella, Fan Deng, and Wolfgang Nejdl. Efficient incremental near duplicate detection based on locality sensitive hashing. In *DEXA*, pages 152–166, 2010.

[FHMN10] Marco Fisichella, Eelco Herder, Ivana Marenzi, and Wolfgang Nejdl. Who are you working with? - visualizing tel research communities -. In *ED-MEDIA: Proceedings of the International Conference on Educational Multimedia, Hypermedia & Telecommunications*, 2010.

[Fie09] Andy Field. *Discovering Statistics Using SPSS (Introducing Statistical Methods)*. Sage Publications Ltd, third edition edition, January 2009.

[FIT11] Oana Frunza, Diana Inkpen, and Thomas Tran. A machine learning approach for identifying disease-treatment relations in short texts. *IEEE Trans. on Knowl. and Data Eng.*, 23(6):801–814, June 2011.

[FKCM12] Marco Fisichella, Ricardo Kawase, Juri Luca De Coi, and Maristella Matera. User profile based activities in flexible processes. In *WIMS: Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, 2012.

[Fli] Flickr. http://www.flickr.com.

[FM11] Marco Fisichella and Maristella Matera. Process flexibility through customizable activities: A mashup-based approach. In *ICDE Workshops*, pages 226–231, 2011.

[FN] Marco Fisichella and Wolfgang Nejdl. Generative model for content analysis of natural language documents. In *TKDE: Under Submission at IEEE Transactions on Knowledge and Data Engineering Journal*.

[FPT06] Marco Fisichella, Alessandra Pandolfi, and Valerio Targon. Risk government in dangerous goods transportation. In *AED: Proceeding of the Advanced Engineering Design Conference*, 2006.

[FPT$^+$07] Marco Fisichella, Alessandra Pandolfi, Valerio Targon, Luciano Raso, and Fabio Siragusa. Dangerous goods governance. In *Multidisciplinarity and innovation, ASP Projects 1, Telesma Edizioni, Lomazzo (Co)*, 2007.

[FSCD11] Marco Fisichella, Avaré Stewart, Alfredo Cuzzocrea, and Kerstin Denecke. Detecting health events on the social web to enable epidemic intelligence. In *SPIRE*, pages 87–103, 2011.

[FSDN10] Marco Fisichella, Avaré Stewart, Kerstin Denecke, and Wolfgang Nejdl. Unsupervised public health event detection for epidemic intelligence. In *CIKM*, pages 1881–1884, 2010.

[FWC$^+$11] Christina Frank, Dirk Werber, Jakob P. Cramer, Mona Askar, Mirko Faber, Matthias an der Heiden, Helen Bernard, Angelika Fruth, Rita Prager, Anke Spode, Maria Wadl, Alexander Zoufaly, Sabine Jordan, Markus J. Kemper, Per Follin, Luise Mller, Lisa A. King, Bettina Rosner, Udo Buchholz, Klaus Stark, and Grard Krause. Epidemic profile of shiga-toxinproducing escherichia coli o104:h4 outbreak in germany. *New England Journal of Medicine*, 365(19):1771–1780, 2011.

[FYYL05] Gabriel Pui Cheong Fung, Jeffrey X. Yu, Philip S. Yu, and Hongjun Lu. Parameter free bursty events detection in text streams. In *Proceedings of the 31st international conference on Very large data bases*, VLDB '05, pages 181–192. VLDB Endowment, 2005.

[Gar79] E. Garfield. Is citation analysis a legitimate evaluation tool? *Scientometrics*, 1:359–375, 1979. 10.1007/BF02019306.

[GHY02] Ralph Grishman, Silja Huttunen, and Roman Yangarber. Information extraction for enhanced access to disease outbreak reports. *J. of Biomedical Informatics*, 35(4):236–246, 2002.

[GIM99] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *Proceedings of 25th International Conference on Very Large Data Bases (VLDB)*, pages 518–529, 1999.

[Gut84] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceedings of Annual Meeting (SIGMOD'84)*, 1984.

[GW05] Like Gao and Xiaoyang Sean Wang. Continuous similarity-based queries on streaming time series. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 17(10):1320–1332, 2005.

[GW07] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice Hall, 2007.

[HCL07] Qi He, Kuiyu Chang, and Ee-Peng Lim. Analyzing feature trajectories for event detection. In *SIGIR*, pages 207–214, 2007.

[HCLZ07] Qi He, Kuiyu Chang, Ee-Peng Lim, and Jun Zhang. Bursty feature representation for clustering text streams. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 491–496, 2007.

[HGEF] Nathalie Henry, Howard Goodell, Niklas Elmqvist, and Jean-Daniel Fekete. 20 years of four hci conferences: A visual exploration.

[HMA09] Parisa Haghani, Sebastian Michel, and Karl Aberer. Distributed similarity search in high dimensions using locality sensitive hashing. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '09, pages 744–755, New York, NY, USA, 2009. ACM.

[Hof99] Thomas Hofmann. Probabilistic latent semantic analysis. In *UAI*, pages 289–296, 1999.

[IM98] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings on Annual ACM Symposium on Theory of Computing (STOC)*, pages 604–613, 1998.

[Ind06] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53:307–323, May 2006.

[JMF99] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, September 1999.

[Jol02] I. T. Jolliffe. Principal component analysis and factor analysis. In *Principal Component Analysis*, Springer Series in Statistics, pages 150–166. Springer New York, 2002. 10.1007/0-387-22440-8-7.

[JS08] Edwin H. Jacox and Hanan Samet. Metric space similarity joins. *ACM Transactions on Database Systems (TODS)*, 33(2), 2008.

[KA04] Giridhar Kumaran and James Allan. Text classification and named entities for new event detection, 2004.

[Kaw10] Noriaki Kawamae. Latent interest-topic model: finding the causal relationships behind dyadic data. In *CIKM*, pages 649–658, 2010.

[KBTea09] Mikaela Keller, Michael Blench, Herman Tolentino, and et al. Use of unstructured event-based reports for global infectious disease surveillance. 15(5), May 2009.

[KFN10] Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. Boilerplate detection using shallow text features. In Brian D. Davison, Torsten Suel, Nick Craswell, and Bing Liu, editors, *WSDM*, pages 441–450. ACM, 2010.

[KL04] Robert Krauthgamer and James R. Lee. Navigating nets: simple algorithms for proximity search. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 798–807, 2004.

[Kle02] Jon M. Kleinberg. Bursty and hierarchical structure in streams. In *KDD*, pages 91–101, 2002.

[KOST04] Nick Koudas, Beng Chin Ooi, Heng Tao Shen, and Anthony K. H. Tung. Ldc: Enabling search by partial distance in a hyper-dimensional space. In *Proceedings of the 20th International Conference on Data Engineering (ICDE)*, pages 6–17, 2004.

[KOT04] Nick Koudas, Beng Chin Ooi, Kian-Lee Tan, and Rui Zhang 0003. Approximate nn queries on streams with guaranteed error/performance bounds. In *VLDB*, pages 804–815, 2004.

[KS97] Norio Katayama and Shin'ichi Satoh. The sr-tree: An index structure for high-dimensional nearest neighbor queries. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 1997.

[KT05] Rob Koper and Colin Tattersall. *Learning design: A handbook on modelling and delivering networked education and training.* Springer, 2005.

[L08] Xiang Lian and Lei Chen 0002. Efficient similarity search over future stream time series. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 20(1):40–54, 2008.

[LAD$^+$02] Victor Lavrenko, James Allan, Edward DeGuzman, Daniel LaFlamme, Veera Pollard, and Stephen Thomas. Relevance models for topic detection and tracking. In *Proceedings of the second international conference on Human Language Technology Research*, pages 115–121, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[LJW$^+$07] Qin Lv, William Josephson, Zhe Wang, Moses Charikar, and Kai Li. Multi-probe lsh: Efficient indexing for high-dimensional similarity search. In *Proceedings of 33th International Conference on Very Large Data Bases (VLDB)*, pages 950–961, 2007.

[LMF$^+$11] Jens P. Linge, Jas Mantero, Flavio Fuart, Jenya Belyaeva, Martin Atkinson, and Erik van der Goot. Tracking media reports on the shiga toxin-producing escherichia coli. In *In Proceedings of the Electronic Healthcare International Conference (eHealth)*. Springer, 2011.

[LMWY01] W. Lam, H. M. L. Meng, K. L. Wong, and J. C. H. Yen. Using contextual analysis for news event detection. *International Journal on Intelligent Systems*, 2001.

[LSF$^+$10] Jens Linge, Ralf Steinberger, Flavio Fuart, Stefano Bucci, Jenya Belyaeva, and Monica Gemo. Medisys: Medical information system. In *Advanced ICTs for Disaster Management and Threat Detection: Collaborative and Distributed Frameworks*, pages 131–142. Ed. Eleana Asimakopoulou and Nik Bessis. Hershey: IGI Global, 2010.

[LWLM05] Zhiwei Li, Bin Wang, Mingjing Li, and Wei-Ying Ma. A probabilistic model for retrospective news event detection. In *SIGIR*, pages 106–113, 2005.

[Mad04] Lawrence C. Madoff. Promed-mail: An early warning system for emerging disease. 2(39):227–232, July 2004.

[Met] MetaMap. http://mmtx.nlm.nih.gov.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schuetze. *Introduction to Information Retrieval*. Cambridge University Press, 1 edition, July 2008.

[MWC10] Zhaoyan Ming, Kai Wang, and Tat-Seng Chua. Prototype hierarchy based clustering for the categorization and navigation of web collections. In *SIGIR*, pages 2–9, 2010.

[MZ05] Qiaozhu Mei and ChengXiang Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 198–207, New York, NY, USA, 2005. ACM.

[NAXC08] Ramesh Nallapati, Amr Ahmed, Eric P. Xing, and William W. Cohen. Joint latent topic models for text and citations. In *KDD*, pages 542–550, 2008.

[NMTM00] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using em. *Mach. Learn.*, 39:103–134, May 2000.

[NSK+10] Katja Niemann, Uta Schwertel, Marco Kalz, Alexander Mikroyannidis, Marco Fisichella, Martin Friedrich, Michele Dicerto, Kyung-Hun Ha, Philipp Holtkamp, and Ricardo Kawase. Skill-based scouting of open management content. In *EC-TEL*, pages 632–637, 2010.

[OMD09] Xavier Ochoa, Gonzalo Mendez, and Erik Duval. Who we are: Analysis of 10 years of the ed-media conference. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications ED-Media 2009*, pages 189–200, 2009.

[Ope] OpenCalais. http://www.opencalais.com.

[Org] World Health Organization. http://www.who.int/csr/don/en/index.html.

[Pan06] Rina Panigrahy. Entropy based nearest neighbor search in high dimensions. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1186–1195, 2006.

[PCKC06] C Paquet, D Coulombier, R Kaiser, and M Ciotti. Epidemic intelligence: a new framework for strengthening disease surveillance in Europe. *Euro Surveillence*, 11(12):212–214, 2006.

[PW05] Ros Paramythis and Stephan Weibelzahl. A decomposition model for the layered evaluation of interactive adaptive systems. In *In Proceedings of UM 2005, LNAI 3538*, pages 438–442. Springer, 2005.

[SA00] Russell Swan and James Allan. Automatic generation of overview timelines. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56. ACM Press, 2000.

[Sam06] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, August 8, 2006.

[SCO] SCORM. http://adlcommunity.net/mod/resource/view.php?id=458.

[SFD10] Avaré Stewart, Marco Fisichella, and Kerstin Denecke. Detecting public health indicators from the web for epidemic intelligence. In *eHealth*, pages 10–17, 2010.

[SFvdG+08] Ralf Steinberger, Flavio Fuart, Erik van der Groot, Clive Best, Peter von Etter, and Roman Yangarber. Text mining from the web for medical intelligence. *Mining Massive Data Sets for Security*, 19:295–310, 2008.

[SG07] Mark Steyvers and Tom Griffiths. *Probabilistic Topic Models*. Lawrence Erlbaum Associates, 2007.

[SYUK00] Yasushi Sakurai, Masatoshi Yoshikawa, Shunsuke Uemura, and Haruhiko Kojima. The a-tree: An index structure for high-dimensional spaces using relative approximation. In *Proceedings of 26th International Conference on Very Large Data Bases (VLDB)*, pages 516–526, 2000.

[TFF07] A. Torralba, R. Fergus, and W. T. Freeman. Tiny images. Technical Report MIT-CSAIL-TR-2007-024, Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, 2007.

[TSK05] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[TYSK10] Yufei Tao, Ke Yi, Cheng Sheng, and Panos Kalnis. Efficient and accurate nearest neighbor and closest pair search in high-dimensional space. *ACM Trans. Database Syst.*, 35:20:1–20:46, July 2010.

[Vla04] Michail Vlachos. Identifying similarities, periodicities and bursts for online search queries. In *In Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 131–142. ACM Press, 2004.

[WM98] Howard D. White and Katherine W. Mccain. Visualizing a discipline: An author co-citation analysis of information science. *Journal of the American Society for Information Science*, 49:1972–1995, 1998.

[WSB98] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of 24th International Conference on Very Large Data Bases (VLDB)*, pages 194–205, 1998.

[WVSss] F. Wild, C. Valentine, and P. Scott. Shifting interests: Changes in the lexical semantics of ed-media. *Journal of e-Learning*, in press.

[Yan06] Roman Yangarber. Verification of facts across document boundaries. *In Proceedings International Workshop on Intelligent Information Access*, 2006.

[YOTJ01] Cui Yu, Beng Chin Ooi, Kian-Lee Tan, and H. V. Jagadish. Indexing the distance: An efficient method to knn processing. In *Proceedings of 27th International Conference on Very Large Data Bases (VLDB)*, 2001.

[You] YouTube. http://www.youtube.com.

[YPC98] Yiming Yang, Tom Pierce, and Jaime Carbonell. A study of retrospective and on-line event detection. In *SIGIR '98: Proceedings of the 21st* *annual international ACM SIGIR conference on Research and development in information retrieval*, pages 28–36, New York, NY, USA, 1998. ACM.

[YZCJ02] Yiming Yang, Jian Zhang, Jaime Carbonell, and Chun Jin. Topic-conditioned novelty detection. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 688–693, New York, NY, USA, 2002. ACM.

[Zol86] V.M. Zolotarev. *One-dimensional stable distributions*. Translations of mathematical monographs. American Mathematical Society, 1986.

[ZZH$^+$09] Duo Zhang, ChengXiang Zhai, Jiawei Han, Ashok Srivastava, and Nikunj Oza. Topic modeling for olap on multidimensional text databases: topic cube and its applications. *Stat. Anal. Data Min.*, 2(5-6):378–395, 2009.

[ZZW07] Kuo Zhang, Juan Zi, and Li G. Wu. New event detection based on indexing-tree and named entity. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 215–222, New York, NY, USA, 2007. ACM.