WHEN IN DOUBT ASK THE CROWD

LEVERAGING COLLECTIVE INTELLIGENCE FOR IMPROVING EVENT DETECTION AND MACHINE LEARNING

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

Doktor der Naturwisschenschaften

Dr. rer. nat.

genehmigte Dissertation von

Dipl.-Ing. Mihai Georgescu

geboren am 27. November 1984, in Rîmnicu Sărat, Rumänien

2015

Referent: Prof. Dr. Wolfgang Nejdl
Ko-Referent: Prof. Dr. Kurt Schneider
Tag der Promotion: 8. Mai 2015

ABSTRACT

The Internet and more specifically Web 2.0 is a promoter and enhancer of collective intelligence as it allows people to easily generate, store and retrieve information that can be shared without difficulty. Thus both the expression and exploitation of the wisdom of the crowds are facilitated by applications relying on collective and collaborative intelligence. The contribution of this thesis in leveraging the wisdom of the crowds effect is twofold. First, we propose methods for extracting event-related information from the collectively contributed Wikipedia, thus helping those who are interested in having a comprehensive overview of a happening to address their doubts about the reported facts. Second, we provide ways of exploiting on-demand requested wisdom of the crowds, by involving crowdsourcing in supervised machine learning. Thus, machines can call the crowd and ask for assistance whenever there are doubts about the tasks that need to be solved.

We first focus on how collaborative intelligence in Wikipedia is manifested in the process of information actualization as a reaction to the new events. Real-world events directly influence the collaborative editing of Wikipedia articles about related entities. Consequently, as new events take place all over the world, Wikipedia users update the articles corresponding to the entities involved in these events, or influenced by them, causing an avalanche of edits on several articles, as more information regarding the event becomes available. The interactions of contributors with the articles give us clues on whether certain updates are event-related or not, or whether concurrent updates are a sign of participation in a common event. We identify and leverage these patterns in order to identify those updates that are a consequence of events and summarize them in a comprehensive way that presents all relevant information, even if intentionally forgotten. Moreover, as events can be defined as relationships between entities at a certain time point caused by a common happening, we investigate how concurrent edits can be used as indicators for entities being involved in common events.

We then concentrate on how machine learning can benefit from crowdsourcing. We first propose methods to aggregate multiple crowd labels in order to produce reliable annotated content that can be used for supervised machine learning. The proposed methods take advantage of the workers' history of already solved tasks in order to simultaneously assess worker expertise and find the underlying hidden labels. Then, we go a step further, and propose to couple active learning with crowdsourcing in an integrated framework. Thus, machines and humans can work together towards improving their performance at a specific task. An automatic algorithm can learn from the crowd how to do its task in an active learning manner. When the algorithm has a doubt about a label, or needs to reduce the doubts over the task at hand, it can directly ask the crowd, and get the reliable labels that it needs in order for it to become better. The proposed integrated framework accounts for different worker expertise, instance selection strategies, as well as various levels of resource allocation.

Keywords: *Collective Intelligence, Event Detection, Crowdsourcing, Active Learning*

ZUSAMMENFASSUNG

Das Internet und insbesondere Web 2.0 fördert und verstärkt kollektive Intelligenz, weil es Menschen erlaubt Informationen einfacher zu erzeugen, zu speichern und abzurufen, die ohne Schwierigkeiten mitgeteilt werden können. Somit wird sowohl der Ausdruck als auch die Verwertung der Weisheit der Massen, mittels Anwendungen die auf kollektiver und kollaborativer Intelligenz basieren, erleichtert. Der Beitrag dieser Arbeit zu dem wirksamen Einsatz des "Weisheit der Massen"-Effekts ist zweifach. Erstens, wir schlagen Methoden zur Extraktion von ereignisbezogenen Informationen aus der kollektiv erstellten Wikipedia vor. Dadurch helfen wir denen, die daran interessiert sind einen umfassenden Überblick über ein Geschehen zu bekommen, um ihre Zweifel an den berichteten Fakten auszurumen. Zweitens, wir zeigen Wege zur bedarfsweisen Nutzung der Weisheit der Massen, indem wir Crowdsourcing für überwachtes maschinelles Lernen einsetzen. Auf diese Weise können Maschinen die Masse aufrufen und um Hilfe bitten, wenn es Zweifel über die Aufgaben, die gelöst werden müssen, gibt.

Wir konzentrieren uns zuerst darauf, wie sich kollaborative Intelligenz in Wikipedia im Vorgang der Informationsaktualisierung als Reaktion auf neue Ereignisse manifestiert. Ereignisse in der realen Welt beeinflussen direkt die kollaborative Bearbeitung von Wikipedia-Artikeln der beteiligten Entitäten. Dies führt zu einer Kette von Änderungen an mehreren Artikeln, sobald neue Informationen über das Geschehen zur Verfügung stehen. Die Interaktionen der Beitragenden mit den Artikeln geben uns Hinweise darauf, ob bestimmte Aktualisierungen ereignisbezogen sind und ob gleichzeitige Aktualisierungen von verschiedenen Artikeln ein Zeichen für die Teilnahme der Entitäten an einem gemeinsamen Ereignis sind. Wir identifizieren und nutzen diese Muster, um die Aktualisierungen, die eine Folge von Ereignissen sind, zu finden und dann so zusammenzufassen, dass alle relevanten Informationen präsentiert werden, selbst dann, wenn sie absichtlich ausgelassen wurden. Außerdem, weil Ereignisse als Beziehungen zwischen Entitäten zu einem bestimmten Zeitpunkt, die von einem gemeinsamen Geschehnis verursacht wurden, definiert werden können, untersuchen wir, wie die Nutzung gleichzeitiger Aktualisierungen als Indikatoren dafür, dass Entitäten an gemeinsamen Ereignissen beteiligt sind, genutzt werden können.

Danach konzentrieren wir uns darauf, wie maschinelles Lernen von Crowdsourcing profitieren kann. Zuerst schlagen wir Methoden vor, um mehrere durch die Masse erstellte Labels zu aggregieren, um zuverlässig markierte Inhalte zu erhalten, die für überwachtes maschinelles Lernen verwendet werden können. Die vorgeschlagenen Methoden nutzen die Chronik bereits gelöster Aufgaben, um gleichzeitig die Kompetenz der Arbeiter zu bewerten und die zugrunde liegenden verborgenen Labels zu finden. Dann gehen wir einen Schritt weiter und schlagen vor, aktives Lernen mit Crowdsourcing in einem integrierten Framework zu koppeln. Auf diese Weise können Maschinen und Menschen zusammen an der gegenseitigen Verbesserung ihrer Leistung bei einer bestimmten Aufgabe arbeiten. Ein automatischer Algorithmus kann aktiv von der Masse lernen, wie er seine Aufgabe lösen soll. Wenn der Algorithmus Zweifel an einem Label hat oder Unsicherheit in Bezug auf die Aufgabe reduzieren muss, kann er direkt die Masse fragen, um zuverlässige Labels zu erhalten, die er benötigt, um seine Leistung zu verbessern. Das vorgeschlagene integrierte Framework berücksichtigt Unterschiede in der Kompetenz der Arbeiter, unterschiedliche Auswahlstrategien für Instanzen sowie verschiedene Levels der Ressourcenallokation.

Schlagwörter: *Kollektive Intelligenz, Ereignisdetektion, Crowdsourcing, Actives Lernen*

FOREWORD

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

In Chapter 3 we describe contributions included in:

- *Extracting Event-Related Information from Article Updates in Wikipedia.* Mihai Georgescu, Nattiya Kanhabua, Daniel Krause, Wolfgang Nejdl, Stefan Siersdorfer. In: ECIR, 2013. [GKK+13]

- *Temporal Summarization of Event-Related Updates in Wikipedia.* Mihai Georgescu, Dang Duc Pham, Nattiya Kanhabua, Sergej Zerr, Stefan Siersdorfer, Wolfgang Nejdl. In: WWW (Companion Volume), 2013. [GPK+13]

Chapter 4 is built upon the work published in:

- *WikipEvent: leveraging Wikipedia Edit History for Event Detection.* Tuan Tran, Andrea Ceroni, Mihai Georgescu, Kaweh Djafari Naini, Marco Fisichella. In: WISE, 2014. [TCG+14]

In Chapter 6 is built upon the work published in:

- *Aggregation of Crowdsourced Labels Based on Worker History.* Mihai Georgescu, Xiaofei Zhu. In: WIMS, 2014. [GZ14]

- *Getting by with a Little Help from the Crowd: Practical Approaches to Social Image Labeling.* Babak Loni, Jonathon Hare, Mihai Georgescu, Michael Riegler, Xiaofei Zhu, Mohamed Morchid, Richard Dufour, Martha Larson. In: CrowdMM, 2014. [LHG+14]

- *L3S at MediaEval 2013 Crowdsourcing for Social Multimedia Task.*Mihai Georgescu, Xiaofei Zhu. In: MediaEval, 2013. [GZ13]

- *Fashion-focused creative commons social dataset.* Babak Loni, Maria Menendez, Mihai Georgescu, Luca Galli, Claudio Massari, Ismail Sengör Altingövde, Davide Martinenghi, Mark Melenhorst, Raynor Vliegendhart, Martha Larson. In: MMSys, 2013. [LMG+13]

Chapter 7 we describe contributions included in:

- *When in Doubt ask the Crowd : Employing Crowdsourcing for Active Learning.* Mihai Georgescu, Dang Duc Pham, Claudiu Firan, Ujwal Gadiraju and Wolfgang Nejdl. In: WIMS, 2014. [GPF+14]
- *Map to humans and reduce error: crowdsourcing for deduplication applied to digital libraries.* Mihai Georgescu, Dang Duc Pham, Claudiu S. Firan, Wolfgang Nejdl, Julien Gaugaz. In: CIKM, 2012. [GPF+12]
- *Social Computing for Libraries Data De-Duplication through the Crowd.* Claudiu S. Firan. Mihai Georgescu, Wolfgang Nejdl. In: 10th International Bielefeld Conference, 2012. [FGN12]

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

- *WikipEvent: Temporal Event Data for the Semantic Web.* Ujwal Gadiraju, Kaweh Djafari Naini, Andrea Ceroni, Mihai Georgescu, Marco Fisichella. In: ISWC Posters and Demos, 2014. [GNC+14]
- *Information Evolution in Wikipedia.* Andrea Ceroni, Mihai Georgescu, Ujwal Gadiraju, Kaweh Djafari Naini, Marco Fisichella. In: OpenSym, 2014. [CGG+14]
- *Getting by with a Little Help from the Crowd: Practical Approaches to Social Image Labeling.* Babak Loni, Jonathon Hare, Mihai Georgescu, Michael Riegler, Xiaofei Zhu, Mohamed Morchid, Richard Dufour, Martha Larson. In: CrowdMM, 2014. [LHG+14]
- *WikipEvent: leveraging Wikipedia Edit History for Event Detection.* Tuan Tran, Andrea Ceroni, Mihai Georgescu, Kaweh Djafari Naini, Marco Fisichella, In: WISE, 2014. [TCG+14]
- *Analysing the Duration of Trending Topics in Twitter Using Wikipedia.* Tuan Tran, Mihai Georgescu, Xiaofei Zhu, Nattiya Kanhabua. In: WebSci, 2014. [TGZK14]
- *An Adaptive Teleportation Random Walk Model for Learning Social Tag Relevance.* Xiaofei Zhu, Mihai Georgescu, Wolfgang Nejdl. In: SIGIR, 2014. [ZNG14]
- *When in Doubt ask the Crowd : Employing Crowdsourcing for Active Learning.* Mihai Georgescu, Dang Duc Pham, Claudiu Firan, Ujwal Gadiraju and Wolfgang Nejdl. In: WIMS, 2014. [GPF+14]

- *Aggregation of Crowdsourced Labels Based on Worker History.* Mihai Georgescu, Xiaofei Zhu. In: WIMS, 2014. [GZ14]
- *Workshops Held at the First AAAI Conference on Human Computation and Crowdsourcing: A Report.* Tatiana Josephy, Matt Lease, Praveen Paritosh, Markus Krause, Mihai Georgescu, Michael Tjalve, Daniela Braga. In: AI Magazine 35(2): 75-78, 2014. [JLP+14]
- *Disco: Workshop on Human and Machine Learning in Games.* Markus Krause, François Bry, Mihai Georgescu. In: HCOMP, 2013. [KBG13]
- *L3S at MediaEval 2013 Crowdsourcing for Social Multimedia Task.*Mihai Georgescu, Xiaofei Zhu In: MediaEval, 2013. [GZ13]
- *Temporal summarization of event-related updates in Wikipedia.* Mihai Georgescu, Dang Duc Pham, Nattiya Kanhabua, Sergej Zerr, Stefan Siersdorfer, Wolfgang Nejdl In: WWW (Companion Volume), 2013. [GPK+13]
- *Extracting Event-Related Information from Article Updates in Wikipedia.* Mihai Georgescu, Nattiya Kanhabua, Daniel Krause, Wolfgang Nejdl, Stefan Siersdorfer In: ECIR, 2013. [GKK+13]
- *Fashion-focused creative commons social dataset.* Babak Loni, Maria Menendez, Mihai Georgescu, Luca Galli, Claudio Massari, Ismail Sengör Altingövde, Davide Martinenghi, Mark Melenhorst, Raynor Vliegendhart, Martha Larson In: MMSys, 2013. [LMG+13]
- *Swarming to rank for recommender systems.* Ernesto Diaz-Aviles, Mihai Georgescu, Wolfgang Nejdl In: RecSys, 2012. [DAGN12]
- *Predicting the Future Impact of News Events.* Julien Gaugaz, Patrick Siehndel, Gianluca Demartini, Tereza Iofciu, Mihai Georgescu, Nicola Henze In: ECIR, 2012. [GSD+12]
- *Map to humans and reduce error: crowdsourcing for deduplication applied to digital libraries.* Mihai Georgescu, Dang Duc Pham, Claudiu S. Firan, Wolfgang Nejdl, Julien Gaugaz. In: CIKM, 2012. [GPF+12]
- *FreeSearch: Literatursuche ohne Hindernisse.* Claudiu S. Firan. Mihai Georgescu, Wolfgang Nejdl, Xinyun Sun In: Konferenz der Deutschen Gesellschaft fr Informationswissenschaft und Informationspraxis (DGI) (poster paper), 2012. [FGNS12]
- *Social Computing for Libraries Data De-Duplication through the Crowd.* Claudiu S. Firan. Mihai Georgescu, Wolfgang Nejdl In: 10th International Bielefeld Conference, 2012. [FGN12]

- *FreeSearch - Literature Search in a Natural Way.* Claudiu S. Firan. Mihai Georgescu, Wolfgang Nejdl, Xinyun Sun. In: HCIR, 2011. [FGNS11]
- *LDA for on-the-fly auto tagging.* Ernesto Diaz-Aviles, Mihai Georgescu, Avaré Stewart, Wolfgang Nejdl. In: RecSys, 2010. [DAGSN10]
- *Bringing order to your photos: event-driven classification of Flickr images based on social knowledge.*Claudiu S. Firan, Mihai Georgescu, Wolfgang Nejdl, Raluca Paiu. In: CIKM, 2010. [FGNP10]
- *An Architecture for Finding Entities on the Web.* Gianluca Demartini, Claudiu S. Firan, Mihai Georgescu, Tereza Iofciu, Ralf Krestel, Wolfgang Nejdl. In: LA-WEB/CLIHC, 2009. [DFG+09]
- *Social Knowledge-Driven Music Hit Prediction.* Kerstin Bischoff, Claudiu S. Firan, Mihai Georgescu, Wolfgang Nejdl, Raluca Paiu. In: ADMA, 2009. [BFG+09]

Contents

List of Figures

1 Introduction

The Web 2.0 revolution introduced new ways of generating, storing, sharing and retrieving information, stimulating collective intelligence and bringing forward new ways of collaboration. Collaborative platforms such as Wikipedia, Yahoo! answers, or Delicious, released the potential of the wisdom of the crowds, by enabling large groups to collaborate in the creation of knowledge that is universally available. Collaborative intelligence thus leads to collective intelligence, whose beneficiaries are not only its contributors but all those that have access and means of interpretation for the knowledge generated in the process. The advent of human computation, spurned by the high connectivity offered by the Internet and the Web, has unleashed another previously unexplored facet of the wisdom of the crowds, namely crowdsourcing. Crowdsourcing uses small financial incentives to motivate large groups of people, with different levels of expertise, into solving tasks that are too hard for machines, but nearly trivial for humans. With the help of crowdsourcing the undertaking of tasks that would have previously been too costly, or would have taken a high amount of time to accomplish, is substantially eased.

When in doubt about certain facts, or needing assistance when solving a particular task, the power of the crowd and its wisdom is an important resource, bringing considerable advantages when correctly used. Wikipedia is already one of the most-up-to-date and accurate encyclopedias available, and it is based on the power of collaborative editing supported by a large community. It is an important source of information for most of us, clearing our doubts whenever we are interested in a certain subject. The knowledge created by the crowd can be not only be used directly by humans, but it can also be leveraged for machine learning. When needing to train a supervised machine learning algorithm, labeled data is always necessary. Crowdsourcing makes this task easier, reducing the costs or alleviating the need of hiring experts. Moreover, labeled data can be gathered in an incremental way, in an active learning manner. Thus, the crowd will help the machines to clear their doubts about how to solve their tasks, by asking humans for assistance whenever needed. Therefore, the wisdom of the crowds, either in its freely manifested form as collective

or collaborative intelligence, or requested and paid for through crowdsourcing, can help alleviate doubts and give new insights into various problems, making humans as well as machines more efficient, by making their access to information and knowledge easier.

In our work we propose methods of exploiting the wisdom of the crowds, and leveraging collective intelligence for solving the specific problems of event detection and gathering of high quality annotated instances for machine learning. We exploit the success of Wikipedia, to investigate individual contributions, and identify information that is related to events. We propose a framework for using crowdsourcing for active learning, and in the same time we tackle the issues that arise from the quality and unknown provenance of crowd workers. We will detail the problems that we solve and the proposed solutions in the following.

1.1 Thesis Framing

In this section we give an overview of the general background and introduce some of the key notions that are addressed in this thesis. More specific background and the respective state-of-the art surveys are given in the specific Related Work sections in each chapter. We start by presenting what the wisdom of the crowds is and give examples of where it is manifested. We continue by describing what collective and collaborative intelligence are and how they can be used to facilitate the exploitation of the wisdom of the crowds. As an example of collaborative intelligence we give a short description of Wikipedia, and introduce our approach that uses the update patterns for event detection. Finally, we give a short introduction into the concepts of human computation and crowdsourcing as another dimension of the wisdom of the crowds, and present our approach for leveraging crowdsourcing for improving machine learning.

Wisdom of the Crowds

The *wisdom of the crowds* is the theory that when it comes to answering a question, the aggregated decisions of a larger group of diverse people can be better, therefore more intelligent, than that of most individuals and even that of any smaller collection of experts.

One of the first studies [Gal07] into the process, and this also has become the classical example of the wisdom of the crowds, is related to the point estimation of a continuous quantity. The statistician Francis Galton conducted an experiment at a county fair in 1906, by asking the crowd (800 fair participants) to estimate the weight of an ox. When the individual guesses were averaged, it could be observed that the average was accurate within 1% of the true weight of the animal, and closer than the estimates of most crowd members, and also closer than estimates made by experts.

The process underlying the wisdom of the crowds effect is discussed in detail by James Surowiecki in his book *The Wisdom of Crowds* [Sur05]. The central thesis of the book is that a diverse collection of independently deciding individuals can often make certain types of decisions and predictions better than individuals, or even experts. According to the author, there are 4 criteria that separate a wise crowd, from an irrational one: diversity of opinion (each member should have private information, no matter how different from the other's), independence (the opinions of the members should not be determined by others around them), decentralization (members should be able to specialize and draw on local knowledge) and aggregation (the presence of a mechanism that can turn private judgements into a collective decision). Extending these principles, [OK08] captures the wisdom of the crowds approach in eight conjectures: "(i) it is possible to describe how people in a group think as a whole, (ii) in some cases, groups are remarkably intelligent and are often smarter than the smartest people in them, (iii) the three conditions for a group to be intelligent are diversity, independence, and decentralization, (iv) the best decisions are a product of disagreement and contest, (v) too much communication can make the group as a whole less intelligent, (vi) information aggregation functionality is needed, (vii) the right information needs to be delivered to the right people in the right place, at the right time, and in the right way, (viii) there is no need to chase the expert."

The Internet facilitates large groups of independent individuals of various expertise and skills to work together in a decentralized way. Whether having a specific goal in mind or not, large groups of people, coordinate themselves to create knowledge that none of the composing individuals would completely hold. In specialized systems the generated information is democratically coordinated, through complex mechanisms of contests and conflict resolution, and its aggregation leads to the creation of knowledge that sometimes exceeds the capabilities of domain experts. One example of such a system is Wikipedia, where volunteers contribute motivated by the goal of creating the most up-to-date, accurate and neutral encyclopedia. Other examples of wisdom of the crowds manifested through collaboration include: systems centered around getting feedback such as question answering portals (Yahoo! Answers[1], Stackoverflow[2]), posting boards (Reddit[3] or Slashdot[4]), list composition websites (Ranker[5]), or online forums. Moreover, the collaboration generated by the open source movement helped introduce collective and collaborative intelligence into software development and other domains. An example where the greater goal is kept out of sight is crowdsourcing, where random workers of various expertise are contributing to solving tasks for a small financial incentive. The call for answers to a specific question can be simulated by doing a targeted crawl of the resources shared in social networks or social media. By targeted data mining, answers to a specific problem that are given unintentionally

[1] answers.yahoo.com
[2] stackoverflow.com
[3] reddit.com
[4] slashdot.com
[5] ranker.com

or unconsciously by a large number of people, can be gathered or inferred. In this way, the wisdom of the crowds can be exploited without the members of the crowd being aware that they are participating in the process. Therefore, the wisdom of the crowds can also be exploited by aggregating data gathered from systems dedicated to sharing of resources, such as photo and video sharing portals (Flickr[6], Youtube[7]), or bookmarking services (Delicious[8], Bibsonomy[9]), portals developed for keeping in touch with peers such as social networks(Facebook[10], Lastfm[11]), portals for sharing a person's thoughts and experiences with everybody else such as blogging platforms or microblogging (Twitter[12]). Moreover, the ability to learn and infer meaningful knowledge from big data, either generated by a large number of users, by the crowd, or by pervasive computing, is becoming a crucial asset for software companies today. This is by no means an exhaustive list of showcases for the wisdom of the crowds, we only intended to show the diversity of applications leveraging it directly or indirectly, and of platforms that can be exploited for extracting it; many other success stories being supported by it. The aggregation of the information produced by the individuals contributing to these systems, either consciously participating in a group having a precise goal in mind, or just expressing their own opinions about something, creates knowledge that can be referred to as wisdom of the crowds.

Collective and Collaborative Intelligence

Collective intelligence builds upon the wisdom of the crowds, but shifts the focus from individual isolated inputs and their aggregation, towards the process of knowledge production through collaboration. The shared intelligence of the group emerges from the collective efforts and collaboration, sometimes motivated by competition, of many diverse individuals. Collective intelligence relies on the involvement of a large group of individuals that generate knowledge by means of communication, and a certain degree of coordination. This is not contrary to the wisdom of the crowds as long as the individuals are diverse enough, and the knowledge generated by the aggregation of their inputs is a product of disagreement and contest, guided by democratic principles.

In his book *Collective intelligence: Mankind's emerging world in cyberspace* [LB99], Pierre Levy defines collective intelligence as:

> "It is a form of universally distributed intelligence, constantly enhanced, coordinated in real time, and resulting in the effective mobilization of skills. I'll add the following indispensable characteristic to this definition:

[6] www.flickr.com
[7] youtube.com
[8] delicious.com
[9] bibsonomy.org
[10] facebook.com
[11] last.fm
[12] twitter.com

> The basis and goal of collective intelligence is mutual recognition and enrichment of individuals rather than the cult of fetishized or hypostatized communities."

In his book *New media: An introduction* [Fle05], Terry Flew argues that through interaction with new media, knowledge easily passes between sources resulting in a form of collective intelligence. The use of interactive new media, particularly the Internet, promotes online interaction and knowledge distribution between users by allowing them to to easily create, store and retrieve information that can be shared without difficulty. Thus, through new media, the Internet is facilitating and promoting collective intelligence.

Collaborative intelligence is that facet of collective intelligence that acknowledges identity. Zan Gill argues on her website dedicated to surveying theoretical work relevant to developing a theory of collaborative intelligence[13] that:

> "Collaborative intelligence shifts from the anonymity of collective intelligence to acknowledged identity, as when individuals participate in social networks. Collaborative intelligence offers a method to transform next generation social networks into problem-solving systems. Diverse, generally non-anonymous, credited, time-stamped input into an interactive system is tagged, preserving a database of the unique knowledge, expertise, and priorities of participants, while offering diverse methods of clustering, searching, and accessing their input. "

The collective intelligence of a large group of diverse individuals fulfils all the criteria of a wise crowd producing useful information. It consists of a distributed group mind solving creative problems, based on the dynamics resulting from the collaboration of people with various knowledge levels and diverse skills, to produce outcomes that are viewed by the community as more effective than what independent action could have produced. The Internet, as a rich, but noisy platform, enables the emergence and evolution of complex collaborative intelligence ecosystems, that can leverage judgments made by millions of people to generate collective knowledge that can be used to produce intelligent answers to a wide variety of questions.

Collective intelligence can be harnessed today by means of crowdsourcing, recommender systems, evolutionary computation, wikis, community question answering systems, the open source movement, and many others. Various data mining algorithms rely on the aggregation of data mined from the social web in order to produce valuable knowledge. For example, collaborative filtering [BHK98], a successful algorithm for recommender systems, is based on collecting and analyzing a large amount of information on users' behaviors, activities or preferences and predicting what users will like based on their similarity to other users. Collaborative intelligence, leverages identity and social networks to further motivate participants. Wikipedia has

[13] http://collaborative-intelligence.org

successfully applied the wiki paradigm to the creation of one of the most appreciated encyclopedias. Stackoverflow[14], a community answering system focused on programming, is one of the main sources of information for programmers needing assistance. The open source movement enables the collaboration of a large number of programmers and has helped introduce diversity in the operating systems world through Linux. Many other success stories exist, where knowledge being generated through collaboration by a large diverse group can be of use to the general public.

Wikipedia as Collaborative Intelligence

Wikipedia, one of the most popular websites on the Internet, in October 2014 ranked sixth in the world by traffic according to Alexa[15], is one of the foremost showcases for collaborative intelligence. In Wikipedia, contributors from all around the world collectively created and are continuously maintaining the world's largest and most up-to-date encyclopedia, with articles of high quality in terms of content, accuracy, and neutrality. It has been shown that the quality of Wikipedia articles is comparable to that of the Encyclopedia Brittanica [Gil05]. Wikipedia has been developed with almost no centralized control, without any financial motivation, by volunteers. It is a showcase for collaborative intelligence as contributors with different perspectives, expertise, and skills voluntarily collaborate to create knowledge about a wide variety of topics.

The dimensions of contribution in Wikipedia can be grasped by looking at its Statistics page[16]. In October 2014, only the English Wikipedia numbered 33,9 million pages, 4,6 million articles being contributed to by 22,7 million users over 738,3 million edits. Of the 129,628 active registered users, only 1,186 have administrative privileges, showing that the bulk of contributions come from the general crowd, although control is in the hands of a few users.

We are interested in exploiting the collaborative behavior that drives Wikipedia. In [KCP+07] the authors evaluate how the provenance of Wikipedia contributors evolved over time. They conclude that unlike in the beginnings, where a core of dedicated, specialized elite users were contributing, nowadays, Wikipedia is mostly contributed by the common users, thus making it a perfect example for what the "wisdom of the crowds" is capable of, and how the power of collaborative intelligence can create value. The wisdom of the crowds effect has been shown to be manifested in Wikipedia in [AMP06]. The process of coordinating this huge crowd of contributors is discussed in more detail in [KK08], by studying how the number of editors and the types of coordination (explicit or implicit) affects the quality of the articles. Moreover, contributing to Wikipedia can be seen as a process of collective memory building. In [FM11b, FM11c] the authors investigate the processes triggered by the

[14] stackoverflow.com
[15] http://www.alexa.com/siteinfo/wikipedia.org
[16] http://en.wikipedia.org/wiki/Special:Statistics

Arab Spring events of 2011 and how the Wikipedia community reacts in order to create and preserve knowledge about this particular event so that it can be remembered in the future. In Section 2.2 we provide more details on research based on Wikipedia.

Contribution

In this thesis we investigate how the reaction of Wikipedia contributors can be exploited in order to identify events. Is the Wikipedia community acting like an intelligent crowd in coordinating itself to update the article pages of entities involved in events? Can we identify patterns in the update behavior of the Wikipedia contributors in the proximity of events? If so, can these patterns be used to extract event-related information? Due to the encyclopedic nature of Wikipedia, which involves a process of consolidation, some of the information and interactions caused by the increased activity in time of events will be lost. Can the recovery of this information bring further insights into the underlying events? We analyze how the community of contributors behind Wikipedia mobilizes itself and acts as an intelligent crowd when events happen. This leads to the update of the articles of the involved entities. Leveraging the behaviour spurned by the outbreak of events, we propose methods for identifying the updates that were caused as an effect of the event, and furthermore summarize them, providing an event-detection algorithm that leverages Wikipedia editing activities. Not only can we detect events in retrospect, that might have been forgotten, but we also discover information that at the time was considered to be important by the crowd, but later on was removed or overly summarized, for the sake of brevity or to maintain the encyclopedic standards of Wikipedia. Moreover, we introduce an online application that, given an entity reports information about the events where it was involved, based on the proposed methods.

Considering the fact that events can be defined based on the entities that interact with each other as either a cause or a consequence of the event, we also investigate how the crowd wisdom can be used to detect events following this definition. To this effect we examine the concurrent updates that affect entities. The intuition behind this approach is that entities involved in the same event should be updated with similar content, or even more specific, mentions of the other entities participating in the considered event have a higher probability to appear in the articles of entities that are also involved in the event. Therefore the crowd would give an indication of which entities have been affected or involved in common events. Wikipedia also provides a dedicated portal where its contributors can insert and curate information about events, organized in a chronological order. We compare our automatic approach of detecting events with this other example of crowd intelligence, and find that they complement each other.

In summary, propose methods for event-detection to shed more light on past events, or to detect and track events as they happen, by leveraging the collaborative editing behavior of Wikipedia contributors, and we show that the task of event

detection can be based on the exploitation of crowd intelligence.

Crowdsourcing

Human computation, when involving large numbers of humans can be seen as a manifestation of collective intelligence, thus a way of exploiting the wisdom of the crowds. A concise definition for it is "a paradigm for utilizing human processing power to solve problems that computers can not yet solve" [VA09a]. A survey and a taxonomy of the field has been given in [QB11]. The authors classify human computation systems based on six distinguishing factors: motivation, human skill, aggregation, quality control, process order, and task request cardinality. Also in [QB11], the authors compile a list of definitions for human computation, of which we reproduce here the ones that fit best to our work:

> "the idea of using human effort to perform tasks that computers cannot yet perform, usually in an enjoyable manner" [LVA09]

> "systems of computers and large numbers of humans that work together in order to solve problems that could not be solved by either computers or humans alone" [QB09]

From all human computation systems, in this thesis we are most interested in crowdsourcing. Crowdsourcing is a term first coined by Jeff Howe in an article in the Wired Magazine [How06]. Howe's website[17] defines crowdsourcing as:

> "Crowdsourcing is the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call."

After studying multiple definitions of crowdsourcing, [EAGLdG12] comes up with an integrated definition:

> "Crowdsourcing is a type of participative online activity in which an individual, an institution, a non-profit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task, of variable complexity and modularity, and in which the crowd should participate bringing their work, money, knowledge and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the crowdsourcer will obtain and utilize to their advantage that what the user has brought to the venture, whose form will depend on the type of activity undertaken."

[17] http://crowdsourcing.typepad.com/

Crowdsourcing can be seen as general model for problem solving that can be explained through the crowd wisdom theory, an exercise of collective intelligence [Bra08]. Crowdsourcing can be seen as collective intelligence, because a large group of individuals is mobilized to solve a specific problem. The diversity of the individuals, motivated in various ways, assures that the wisdom of the crowds will come into play. Viewed at the low level of an individual in the crowd, it is only outsourcing of tasks towards unknown less paid workers, but usually each task is a part of a more complex system, and all the inputs from the crowd are aggregated to create superior knowledge. Similar small tasks and their aggregation are equivalent to the classical example of wisdom of the crowds, where participants of a fair were asked to estimate the weight of a bull. To understand that collective intelligence is at work here, we have to take the role of employer of the crowd, and having an overview of the ensemble of small tasks, the higher goal will come into view.

A survey of the available crowdsourcing systems in the world-wide web is given in [DRH11]. The survey identifies the key challenges of crowdsourcing systems: how to recruit contributors, what they can do, how to combine their contributions and how to manage abuse. They classify the existing crowdsourcing systems based on the nature of collaboration and the type of target problem to be solved; further distinction can be made based on dimensions of the key challenges. In Section 5.2 we provide more details about research on crowdsourcing that is related to this thesis.

Contribution

In this thesis we focus on what has become the traditional way of using crowdsourcing, through a microtask marketplace. In such marketplaces, the party that needs work being done is called a requester, and those that fulfil it are called workers. The requester posts a microtask that can be solved in a short time, for a monetary reward, usually small. Workers access the marketplace, and according to their skill or preferences they choose the microtasks to solve. The requesters get their work done for a fraction of the money they would need if hiring experts, the workers get financially rewarded, and sometimes also entertained.

Crowdsourcing can be used for a large variety of tasks, ranging from labeling if objects appear in images, to more complex tasks such as translations of text. Crowdsourcing has net advantages over employing experts for solving tasks. On the one hand, from a financial point of view, it can be much more cost-efficient. On the other hand ,due to the large availability of crowd-workers, it can be much faster. Nevertheless, this comes at the cost of quality. The more you pay the workers, the higher the quality of the work produced by them is, but this will lower the advantage of cost-effectiveness, and leave only the time advantage. Even so, methods have been developed to keep the costs low while at the same time keeping the quality standards high.

One application of crowdsourcing is the gathering of labels to be used for super-

vised learning. Thus, machines can learn from humans at a very low cost. Supervised machine learning is of course sensitive to the quality of the labels. This problem is accentuated when the labels are produced by the crowd. One of the most common strategies for addressing these problems is to involve redundancy, and get more than one label for each item, and then use an aggregation mechanism to find the true underlying label. To this end, we propose methods for aggregating multiple crowd produced labels, in order have high quality labels that can be used for machine learning, or even for active learning. Active learning is a special case of semi-supervised machine learning in which a learning algorithm is able to interactively query an information source to obtain the desired outputs at new data points [Set10]. If the information source for gathering labels is crowdsourcing, machines and humans can work together towards reciprocally improving their performances on a specific task. We propose methods to employ crowdsourcing for active learning, in a system where humans and an automatic supervised machine learning algorithm work together in a symbiotic way. Thus, the machines will ask the humans for help, when they are in doubt, or in trouble, for new examples that when labeled would lead to a better performance. In this thesis we address the following questions. Can machine learning be assisted by crowdsourcing? Can we provide an algorithm for the aggregation of multiple crowd-provided labels that is better than the simple majority voting? How can we efficiently couple active learning with crowdsourcing, such that a machine can autonomously learn by requesting labeled items on demand? What are the challenges raised by such a system and how can we overcome them?

In summary,we investigate how crowdsourcing, when done correctly with provisions for tackling the quality problem, can benefit machine learning. We propose methods for quality assurance and worker reliability assessments, and a framework for employing crowdsourcing for active learning.

1.2 Thesis Structure

In the first part of the thesis we exploit the wisdom of the crowds and its manifestation on the collaborative platform Wikipedia, by leveraging the collaboration patterns that make Wikipedia so successfull, for event detection. In Chapter 2 we motivate our approaches for the exploitation of collaborative intelligence from Wikipedia edit activities, and survey the related work. We analyze the behavior of the crowd that supports Wikipedia and the process of updating articles when their corresponding entities are involved in, or affected by events. We present two approaches that go beyond the state-of the art by exploiting the Wikipedia Edit History for event detection. In Chapter 3, we propose algorithms to identify those updates that are consequences of events and summarize them in a comprehensive way that identifies all relevant information, even if intentionally forgotten. Thus, we are able to recover the individual contributions from the process of mutual agreement towards a neutral and brief description of an event. Therefore, we are able to offer a comprehensive view over the

event, that encompasses diverse opinions and points of view. In Chapter 4 we propose algorithms to identify events as relationships between entities by leveraging the concurrent editing behaviour for entities that are related because of an event. This enables us to track the evolution of entities and their relationships as they evolve through involvement in events. Finally, we compare the proposed event-extraction methods with a crowd managed event portal, showing their complementarity.

In the second part of the thesis we investigate a more direct way of exploiting the wisdom of the crowds, namely crowdsourcing. Crowdsourcing by the means of micro-task platforms such as Amazon's Mechanical Turk[18] or Crowdflower[19] has become the standard in recent years. Chapter 5 starts by introducing some of the problems that arise when employing on-demand requested collective intelligence as crowdsourcing, and surveys the related work. We study how crowdsourcing can be used for generating a high quality ground truth to be used in supervised machine learning algorithms, and how crowdsourcing can be integrated in an active learning framework where humans and machines collaborate. In Chapter 6 we propose a general framework for the aggregation of multiple crowd provided labels, while tackling their noisy nature. The proposed methods simultaneously evaluate worker expertise and reliability, and find the underlying ground truth labels for a set of items. We evaluate our methods on various datasets proving their effectiveness. In Chapter 7, we propose methods for employing crowdsourcing for active learning, as an efficient way to gather labeled instances. We explore the challenges created by employing the proposed methods, such as the crowd label quality issue and the various resource allocation schemes and selection strategies. In order to prove the effectiveness of the proposed methods of combining active learning with crowdsourcing we apply them to the task of deduplication of scientific publications. Furthermore, we integrate and test our methods in a live publication search system.

Finally, Chapter 8 concludes the thesis with an enumeration of the contributions, while also discussing possible future research directions and open challenges.

[18] www.mturk.com
[19] www.crowdflower.com

1.3 Contributions of this Thesis

Our various contributions to the exploitation of the wisdom of the crowds can be summarized as follows:

- we analyze the behavior of the crowd that supports Wikipedia, in its collaboration on updating articles when their corresponding entities are involved in or affected by events
- we develop algorithms to identify those updates that are a consequence of events and summarize them in a comprehensive way that identifies all relevant information, even if intentionally forgotten
- we develop algorithms to identify events as relationships between entities by leveraging the concurrent editing behaviour for entities that are related because of events
- we propose a general framework for the aggregation of multiple crowd provided labels to be used in machine learning. The proposed methods simultaneously evaluate worker expertise and reliability, and find the underlying ground truth labels for a set of items.
- we introduce methods for employing crowdsourcing for active learning, and explore challenges arisen by employing such a system, taking into consideration the crowd label quality issue and experimenting on diverse resource allocation schemes and selection strategies.
- we apply the proposed methods for active learning with crowdsourcing for the task of entity deduplication in a live publication search system

2 Leveraging Manifested Collaborative Intelligence for Event Detection

2.1 Motivation

Wikipedia is a free multilingual online encyclopedia covering a wide range of general and specific knowledge in about 33.1 million articles (∼4.7 million in the English version). It is continuously kept up-to-date and extended by a community of over 1.9 million contributors, with an average of 10.4 million edits *per month* observed in 2014.[1]. The dimensions of collaboration exhibited in Wikipedia make it one of the foremost examples of a social network manifesting distributed collaborative intelligence.

One of the reasons that drives editing and updating in Wikipedia is the occurrence of new events in the real world such as elections, accidents, political conflicts, or sport events. In the context of a political argument between the US president Obama and the republican Joe Wilson, which immediately lead to a burst of edits and discussions in Wikipedia, the New York Times wrote: "If journalism is the first draft of history, what is a Wikipedia entry when it is updated within minutes of an event to reflect changes in a person's biography?"[2].

An event is something that happens at a certain time in a specific place. Traditional event detection methods use collections of web articles as a data source, and either cluster articles based on semantics and timestamps, or exploit the distribution of entities over articles in time, in order to identify events. We tackle this problem by using Wikipedia as a collaboratively contributed source of versioned documents, and exploit the apparent update patterns. Contrary to using web documents as a datasource, in Wikipedia knowledge about events and their evolution as well as of the participating entities is constructed incrementally. Compared to a static collection

[1] http://stats.wikimedia.org

[2] http://bits.blogs.nytimes.com/2009/09/10/the-wikipedia-battle-over-joe-wilsons-obama-heckling/

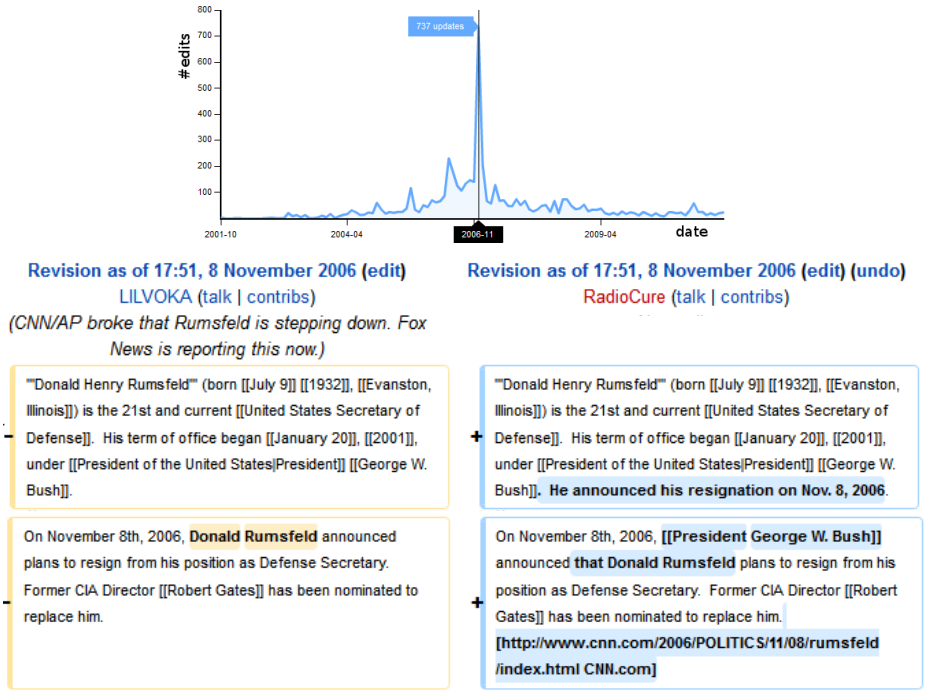


Figure 2.1 On November 8, 2006, the resignation from the U.S. Secretary of Defense of Donald Rumsfeld, caused a burst of updates. Two event-related updates are shown, and contributors, timestamps, comments, and the differences of two revisions highlighted.

of documents, Wikipedia allows us to investigate the evolution of entities as they are affected by, or directly involved in events, and how this is reflected in the behavior of the community. Compared to other social media, like Twitter or Youtube, Wikipedia has some advantages such as: (i) it provides information with higher reliability and quality, because of its goal of being the most accurate and neutral encyclopedia, (ii) the produced information is organized naturally around entities, (iii) all the interactions are stored in an Edit History. The interaction of contributors with the articles, give us clues on whether certain updates are event-related or not, or whether concurrent updates are a sign of participation in a common event. We identify and leverage these patterns in order to detect events.

Wikipedia is widely regarded as the most up-to-date encyclopedia available for free. As an encyclopedia, Wikipedia covers much of what is of importance for a general reader. As it does not have periodical releases, and is a live encyclopedia, Wikipedia has to be constantly updated when events happen. Real-world events directly influence the collaborative editing of Wikipedia articles about entities involved in the events. Consequently, as new events take place all over the world, Wikipedia users will update the articles corresponding to the entities involved in these events, or influenced by them, causing an avalanche of edits on several articles, as more information regarding the event becomes available. Real-life events like new scientific

findings, resignations, deaths, or catastrophes serve as triggers for collaborative editing of articles about affected entities such as persons or countries. As an example of the reaction to a real event, that can be observed in Wikipedia, Figure 2.1 shows typical updates as well as a plot depicting the burst of edits triggered by Donald Rumsfeld's resignation in November 8, 2006. Because Wikipedia has well defined standards for the quality of its articles, and it strives to provide a point of view as neutral as possible, in the course of the event, through a mutual agreement process the different points of view converge to a view accepted by the community of contributors. The large community of users is always mobilized and takes care of keeping the accuracy and actuality of the information. To make sense of all the collaborative editing involved, all revisions of an article are kept, in a publicly available Edit History. We propose to leverage this resource as a news source and extract meaningful events from it.

The enormous volume and the fairly reliable quality of information makes Wikipedia a popular source in several research topics. Research utilizing Wikipedia has attracted a large spectrum of interest over the past decade, including knowledge discovery and management, natural language processing, social behaviour study, information retrieval, etc. Much of existing work considers Wikipedia as a static collection, i.e. information once stored is stable or rarely changed over time. However, in practice, Wikipedia grows very rapidly (from 17 millions articles in 250 languages in 2011 to 30 millions articles in 2013 [3]), with new articles published and edited everyday by a large community of active contributors worldwide. This calls for an effective way to analyze and extract information from Wikipedia, taking into account the temporal dynamics.

How many updates in Wikipedia are related to events? Is there a connection between bursts of edits and real-life events? Are there indicators for event-related updates in the textual content and meta annotations of the Wikipedia edits? Can we automatically detect event-related updates? Can we extract relationships that are generated by these events? Can we observe the evolution of such relationships as consequences of events? These are some of the questions we investigate in this thesis by analyzing Wikipedia's publicly available Edit History.

We propose and discuss at length two methods of detecting events based on the Wikipedia Edit History. In Chapter 3 we present a method based on firstly identifying event-related updates by employing a classifier and then summarizing these updates by clustering them. In Chapter 4 we introduce another method of identifying events as dynamic relationships between entities. Hereafter we motivate and briefly introduce the proposed methods for event identification from Wikipedia article updates.

In Chapter 3 we propose to extract event-related information from Wikipedia edits for a given entity and its corresponding article, by first identifying event-related updates and then by clustering these updates in order to map them to their corre-

[3] http://en.wikipedia.org/wiki/Wikipedia

sponding events and to generate summaries. For detecting event-related updates we make use of a combination of filters and classifiers based on burst detection, temporal information, and textual content. The stream of thus detected event-related updates serves as a starting point for identifying the events themselves and creating a meaningful summarization. Our experiments on event extraction, clustering, and summarization show promising results towards generating entity-specific news tickers and timelines. Based on the proposed methods for event-related update identification we present the *Wikipedia Edit Reporter*, a system that supports the exploration of event-related information in Wikipedia over time. For a given entity, our system automatically identifies event-related updates by analyzing the Edit History of the corresponding Wikipedia article. The system generates a meaningful temporal summarization from event-related updates and automatically annotates events in a timeline. Wikipedia Edit Reporter, provides the user with the ability to visualize all the historical information, giving the user a comprehensive view over the evolution of the event, and not only the socially accepted final interpretation of the event.

In Chapter 4 we propose to extract from Wikipedia complex event structures, consisting of a set of entities that are connected at a given time period. The approach copes with the temporal evolution of information, as new events are reported or existing events evolve from time to time. Most current approaches in information extraction assume the static nature of relationships on fixed collections and knowledge bases. Recent attempts have tried to extract structured information from different sources in Wikipedia articles (categories and infoboxes [TEPW11], free texts [KW12a], etc.), but ignored the temporal dynamics of articles in Wikipedia. They assume a predefined or limited schema of the events detected. Other works exploited timely features of articles such as view counts over time [CN10] or edit history [GKK+13]. However, in collaborative environments such as Wikipedia, information and structures are highly dynamic over time. Therefore, we introduce a new method to extract complex event structures from Wikipedia. We propose a new model to represent events by engaging multiple entities correlating in the time dimension. We exploit the Edit History in Wikipedia, which covers a full evolution of articles' content over a long time period. Our method is agnostic to any language constraints, therefore it can be applied to an arbitrary language, and it does not depend on the number of entities to be known a-priori. In principle, our method can detect events from simple schema (such as the release of *new movies*) to complex ones (such as a *revolution*). Our method works naturally with the dynamics of information in Wikipedia; thus, it is able to detect several events pertaining to a number of articles, as soon as the articles' contents change over time. The evolution of an event is captured effectively based on analyzing the user edits history in Wikipedia. Our work provides a foundation for a novel class of evolution-aware entity-based enrichment algorithms, and considerably increases the quality of entity accessibility and temporal retrieval for Wikipedia. We formalize this problem and introduce an efficient end-to-end workflow as a solution.

2.2 Related Work

Temporal Retrieval and Event Detection

Information retrieval models and algorithms have been extremely successful during the last 20 years, providing easy access to all information available on the Web. Only lately more work has been spent on issues related to temporal retrieval, exploration, and analysis of large temporal collection like Web Archives [ASBYG11]. In the area of temporal retrieval, previous works [EG11, KN12] have shown that leveraging the time dimension in ranking can improve the retrieval effectiveness for temporal queries. However, existing work in temporal search does not focus on entities and events. In our investigation, we circumvent the need for Web Archives by using Wikipedia. Its link structure has been shown to be a good indicator for the historical influence of people [TOY+11], considering that all versions of Wikipedia are stored in an Edit History easily accessible. Thus, we can exploit the time dimension to identify historical events in the collaboratively edited version content of Wikipedia.

Event detection has been applied in many contexts including topic detection and tracking [APL98, HCL07, LWLM05], tracking of natural disasters [SOM10], and event-based epidemic intelligence [AMM11, FSDN10].

An event is defined as *something* that happens *at a particular time and space*. This a very abstract, yet powerful definition, that is leveraged by most event-detection methods in various ways. The event-detection problem as recognized in the literature can be defined as follows: "considering a set of documents where each document is associated to an (unknown) event, the goal is to partition this set of documents into clusters such that each cluster corresponds to all documents associated to an event". Therefore, the *something* is characterized by the set(cluster) of documents associated to the event. The holy grail in this body of work has been to automatically acquire a landscape view of a web collection, which answers in a compact manner the questions of: "What Happened?" and "What is New?".

Event detection has been applied so far to text streams such as broadcast news streams or email or conference titles, to click-through data or to social media such as twitter or other resources that benefit from tags. From the point of view of the time relation between the detected events and the detection time, the approaches can be divided into : *Retrospective Event Detection (RED)*: the discovery of previously unidentified events in an accumulated collection and *On Line Event Detection*: the identification of the onset of new events from live feeds in real-time. For Retrospective Event Detection we can distinguish between: *document-pivot approaches*, where the detection of events is done by clustering documents based on semantics and timestamps, and *feature-pivot approaches*, where firstly the temporal and document distributions of words are studied and events of words are discovered.

TDT (topic detection and tracking) was an initiative started in 1998 and intended to develop core technologies for a news understanding systems. A book, "Topic

Detection and Tracking - Event based organization of documents" [All02], containing all the findings was published by James Allan in 2002. This line of research tries to solve 2 problems: *new event detection*, focusing on identifying those news stories that discuss an event that has not been reported earlier, and *event tracking*, where starting from a few sample stories that discuss an event the task is to find all subsequent stories that discuss the same event. One of the proposed solutions for TDT, [APL98], is a single pass clustering algorithm, that clusters stories, and creates new clusters if the content similarity threshold is not met. These clusters of stories represent events. Also using a document-pivot approach, [YPC98] is taking advantage of the temporal distribution of document clusters. Noticing that events are associated with news bursts they do retrospective and on-line event detection either using a modified Group Average Clustering algorithm, or an improved incremental single pass clustering.

[Kle02] identifies a fundamental problem in text data mining: extraction of meaningful structure from document streams that arrive continually over time. The appearance of a topic in a document stream is signaled by a "burst" of activity in certain features rising sharply in frequency as the topic emerges. The stream is modelled as an infinite state automaton, and from the sequence of inter-arrival times an optimal state sequence, that reveals a latent hierarchical structure, is discovered. The bursts are associated with state transitions, and the hierarchy helps at identifying bursts in bursts. [LWLM05] propose a probabilistic model for event detection that incorporates both content and time information. In the model, events are latent variables and articles are observations. The article content in terms of persons, locations, keywords is modeled as bag of words following a multinomial distribution, and time is modeled by a Gaussian Mixture model. Then a generative model is used to identify the events. [FSDN10] employs a similar method for identifying medical events. Using partially supervised text classification, [FYYL05] identifies hot bursty events as a minimal set of bursty features that occur together in certain time windows with strong support of documents in the text stream. Leveraging click-through data, [ZLBM06] identify events as clusters of (query, page) pairs, using a normalized graph cut algorithm. By representing word occurrences as signals and appling signal processing techniques, [HCL07], proposed to analyze feature trajectories in order to identify events. While most of the previous approaches deal with text streams, algorithms that detect events in social media such as Flickr photos, can benefit from metadata such as of tags, or the time and space dimensions (as photos make it readily available via GPS coordinates). Considering events as a single segment of time over which a single activity was taking place, providing a coherent, unifying context, [RGN07] first identify event and place tags, and then use a scale-structure identification for events detection. Using a wavelet transform, [CR09], identifies events in Flickr, by detecting event-related tags, clustering them, and then presenting the photos associated to the tags and events. Leveraging the behavior of Flickr groups related to events, [FGNP10], propose methods for organizing photos naturally, by detecting the events where they were taken. A large body of work is dedicated detecting events by using the Twitter platform,

where users contribute in real-time short messages that are timestamped and sometimes have a location associated to them. In one of the most notable works [SOM10] develop an application for earthquake detection, using tweets as sensors.

Previous work has focused on detecting events from unstructured text like news, using features such as key words or named entities. In Chapter 3, we employ Wikipedia article updates for event detection instead of using traditional news streams. We show that crowd behavior of editing provides strong indicators for events, and enables focused detection of events connected to a *particular entity* by analyzing the corresponding Wikipedia article.

In Chapter 4 we propose to identify events by discovering and verifying dynamic connections among entities that are coupled at a given time period. We formalize this problem and introduce an end-to-end pipeline as a solution. The problem of discovering dynamic relationships and events has been investigated in [DSJY11]. Given a time period and a set of entities, the authors introduce a two-phased approach to identify entity relationships and events. The first phase, named *implicit relationships identification*, is related to the detection of entity pairs that are temporally connected. Query logs are used to estimate how much a given entity is queried over time, as well as to detect those entity pairs that present a peak of queries at the same time. In practice, if two entities have a peak at the same time, then this could be a signal that these two entities have good semantic connection strength, i.e. a strong implicit relationship. False positive pairs, i.e. those sharing a peak without being connected, are then reduced by computing their implicit relatedness using an entity semantic similarity, i.e. point-wise mutual information (PMI). The second phase deals with the identification of dynamic events from the pairs of related entities. Dynamic relationships between entities are represented as a graph and event identification algorithms are then applied. Our work differs from the aforesaid approach in the way that: (i) our work leverages Wikipedia, its edits history, and its temporal dimension; (ii) we describe entities in terms of users' edits over time, considering the corresponding Wikipedia article instead of queries they are involved in; (iii) we propose several classes of entity similarity measures to estimate the confidence of each *implicit relationship*, in addition to the adapted PMI for our domain; (iv) we introduce the *Explicit Relationships Identification* approach; (v) finally, our entity set is not restricted to a specific class, but comprehends several classes, i.e. according to YAGO2 ontology, and is flexible for future extension.

Mining the Wikipedia Edit History

There has been a large amount of research done on Wikipedia. A survey [MMLW09] categorizes and presents the different areas of research to which Wikipedia is relevant, depending on what perspective we see Wikipedia in: an encyclopedia, a corpus, a thesaurus, a database, an ontology, or a network structure. Furthermore, the survey categorizes the different areas that mine meaning from Wikipedia into: solving

natural processing tasks, information retrieval, information extraction, and ontology creation.To this large body of research we add work that mines events from Wikipedia, by exploiting the patterns manifested in the Edit History as a result of events.

Wikipedia is backed by a large number of contributors, that use the wiki paradigm to collaborate towards achieving the declared goals of being the most accurate and up-to-date encyclopedia. The availability of the whole Edit History has enabled researchers to study the community and the way its members interact. The componence of the Wikipedia contributors and its evolution has been studied in [KCP+07], concluding that the success of Wikipedia is coming from a *wisdom of the crowds* type of effect instead of a core group of users with administrative privileges. After an initial period in which contributions usually were coming from registered users with a high level of participation, in recent years Wikipedia receives most contributions from users with low level of participation. Not only the main contributions come from a different category of contributors, but also the user activities have switched focus as Wikipedia evolved. In [KSPC07] the authors show that conflicts and administrative actions have an increased percentage of the entire activity in Wikipedia. [OGB07] also analyzes the Wikipedia community, confirming the findings of [KCP+07], and extending their methodology, provide the WikiXray tool, to identify which fraction of authors are producing most of the changes in Wikipedia's articles, and how the behaviour of these authors evolves over time. From a sociological point of view, in [Mas11] the authors analyzed conversations happening on user talk pages of wikis from a Social Network Analysis perspective. They find that employing the Edit History of these pages is a more accurate way of encoding communication, and extracting social networks. The following assumption from social science is studied in Wikipedia in [CCH+08]. People tend to have attributes that are similar to those of their friend and neighbors because of two distinct reasons: the process of social influence (leading people to adopt behaviors exhibited by those they interact with) and the process of selection (leading people to form relationships whit others who are already similar to them). The findings in Wikipedia pointed to the fact that similarity between two users can serve as an indicator of future interaction, and after this interaction mediated by the increasing similarity, the similarity between the users continues to increase steadily, but at a slower rate, for long periods after initial interactions. In our work, we analyze the Wikipedia community and identify patterns of activity in the editing behavior related to the occurrence of events. We leverage these patterns in order to identify and extract event-related information.

Regarding the behavior of the contributors, a large body of work has been dedicated to studying the patterns of conflict resolution and cooperation. [LL09] shows that a significant number of Wikipedia contributors exhibits selectivity and geographic locality in the pages they edit. By representing Wikipedia's revision history as a temporal, bipartite network with multiple node and edge types for users and revisions, [JL12c] identifies author interactions as network motifs and show how the motif types capture important, diverse editing behaviors. The discovered motifs

are employed for classifying pages as combative or cooperative page or for analyzing trends in the dynamics of editor behavior to explain Wikipedia's content growth.

A well established way of studying the Wikipedia community is to first constructing an edit graph that encodes the user interactions, and then to use it to investigate different aspects. Following this methodology there are several lines of work with different focuses: identifying conflict and controversy [VLS+08, SRMRB12, LDS12, KSPC07, BKLvR09, JSL09], network analysis [Mas11, CCH+08] , trust [AdA07, ACdA+08, FBA10] and contribution quality [WP09, SH07, KK08].

[VLS+08] build a graph from the Edit History where the vertices represent users and edges represent interactions with weights quantifying the dispute, as number of words deleted from each other. Using this graph, the authors identify articles that attract disputes between the contributors, by employing a mutual reinforcement principle to compute iteratively the article controversy and contributor controversy. The controversy in an article is measured by the amount of disputes occurring in articles and the degree of controversy in each dispute. The models are designed based on the premise that an article is more controversial if more disputes are from the less controversial contributors while a contributor is more controversial if he invites more disputes in less controversial articles. Such a model implicitly assumes that the source of controversial articles is inherently the nature of the individual contributors, rather than the subject matter of the articles. Using the proposed models they compute controversy values of the articles and rank them and then they compare the ranking, to a ground truth obtained from the article's dispute tags. [KSPC07] defines a controversy score (controversy revision count) and use a regression model to predict its value. The authors examine conflict expressed by means of reciprocal reverts, using a graph of reverts and identifying clusters of users. A case study is provided where they manually find the affiliation of the identified groups, uncovering poles of opinion. The authors also investigated a set of page metrics including revisions, the length of content, number of contributors etc. as the features of Wikipedia articles to train a support vector machine classifier. The experiments showed that the learnt classifier was able to rank the controversial articles consistent with their actual degree of controversy. Also, they demonstrated the use of visualization in making sense of disputes between users. [SRMRB12] proposes a method to predict the attitude (positive or negative) between two editors based on the Edit History of their interactions, by building a signed network of all editors of an article that also allows to infer whether or not the said article contains controversial material. [BL08] developed a visualization tool which reveals the dominant authors that are most involved in a controversy and who plays what role in the article building process. Subsequently, in [BKLvR09] the same authors employed an edit network derived from the Edit History to illustrate the collaborative work of contributors in Wikipedia. They analyzed the interaction of the contributors in an article to characterize the role each individual user plays during article writing by focusing on one article and computing network indicators that reveal some information about the article itself. The edit network of a

page is defined with weights for nodes(contributors) and edges(interactions between contributors). Then a partition of the graph that has the highest weight between the 2 groups is computed, for which indicators such as bipolarity are proposed as a characteristic of controversy. A visualization for the discovered groups contributes to the better understanding of the article's nature. Along the same line of identifying conflicting groups, [JSL09] investigates a subset of Wikipedia formed from 3 levels into the Physics and Philosophy categories. By identifying adjacent bicliques in the bipartite graph formed by authors and articles, the authors find related controversial topics. Also investigating controversy in Wikipedia, in [LDS12] by analyzing the logs of Edit History, the authors observed that individual contributors only edit a relatively small number of articles, showing that people have only focused expertise and/or interest areas with respect to the areas covered by the entire Wikipedia. Based on this observation, the expert-based similarity is proposed, to evaluate the relevance of articles among each other, and use it together with other standard similarity measures, to discern the influence and impact of several factors which are believed to generate controversies in Wikipedia articles, concluding that controversies arise from specific content typically confined to individual articles themselves, and not to social interaction.

Another use for the Wikipedia Edit History is for assessing the trust of its content and of the contributors. In [AdA07] a content-driven reputation system for Wikipedia authors is introduced, based on the Edit History of one article. Therefore, the quality of a contribution can be predicted just based on the author's computed reputation, measured in terms of text life and edit life. The reputation thus computed is useful as a guide to the value of fresh contributions, which have not yet been vetted by other users. Building up on the previous work, in [ACdA+08] the same authors developed the WikiTrust system, to assign a trust value to each word in an article considering the edit-history of the article and the reputation of the original author of the word and of the authors who edited text near the word. The proposed algorithm takes every revision of an article and does two steps: computes the trust of the edited text, and adjusts the trust of the un-edited text depending on the reputation of the author and the attention paid to it. To validate the algorithm the authors show that text labeled as low trust has a much higher probability of being edited as text with high trust, showing a correlation between trust and future text stability in the hypothesis that correct text is less likely to be revised. In [FBA10] the authors propose a method for categorizing and presenting edits with a measure of significance of each individual editor's contributions. This is done by means of a pipeline consisting of a lexical analyzer, a text difference engine, an action categorizer, and a history summarizer.

Trust is directly related to article quality. Similar methods leveraging the Edit History of an article have been employed in order to asses its quality. [WP09] use an editing distance based on the Edit History of a Wikipedia article to define the transient and persistent contribution. Based on lifecycles the authors define aggregated metrics that serve as features in classifying the articles as high or low quality. The authors

conclude that high quality articles are generally more intensively edited and pass through a stage of extremely high editing before they go through evaluation and become good or featured articles. [SH07] investigate the German Wikipedia and find out that even though there are no contributors specialized in writing high quality articles, the reputation of the contributors is more important than the number of contributors in regard to the quality of articles. The process of coordination that leads to high quality articles is studied in [KK08]. A distinction is made between explicit coordination, in which editors plan the article through communication, and implicit coordination, in which a subset of editors set direction by doing the majority of the work. The authors observed that adding more editors to an article improved article quality only when they used appropriate coordination techniques and was harmful when they did not, demonstrating the critical importance of coordination in effectively harnessing the *wisdom of the crowd* in online production environments such as Wikipedia.

Collaboration patterns become apparent also through the use dedicated visualization tools. We present some of the visualzations that have been proposed in the literature for exposing the inner workings of the collaboration process of Wikipedia. Some of the most notable visualization tools are HistoryFlow [VWD04], RevertGraph [SCPK07], WikinetViz [LDLD08], WikiDashboard [SCKP08], and WikiChanges [NRD08].

HistoryFlow [VWD04] enables the visualization of how article contents evolve through edit histories highlighting patterns of contributors' edit behaviors. This technique reveals some of the patterns that have emerged within Wikipedia: its surprisingly effective self-healing capabilities, the variety of negotiation processes used in reaching consensus; the diversity of authorship, the bursty rhythms of page editing, and the constant change in page contents. In turn, these facts point to some of the key social mechanisms of the community: the importance of having forums for resolving conflicts and the value of fast, efficient notification of changes to aid surveillance. Complementary to HistoryFlow, [Sab07] proposed an adoption coefficient, which indicates the similarity between two corresponding article revisions, to build a tree structure which reflects the evolution of the article, together with the editing activities among contributors. The tree structure reflects actual evolution of page content, revealing reverts, vandalism, and edit wars, which is demonstrated on Wikipedia examples. Orthogonal to the HistoryFlow approach which focuses on content evolution, [KSPC07] built a revert graph to discover conflicts among contributors. They also proposed a supervised classification method to automatically identify controversial articles. [BL08] offers a more general approach to analyze disagreements among Wikipedia contributors by constructing the revision network, since reverts are not the only and the best indicators of conflicts. They also proposed a spectral layout method to visualize conflicts among contributors. WikinetViz [LDLD08] helps visualize and analyze disputes among users in a dispute-induced social network. Each user (and article) are also assigned a controversy score proportional to the amount

of disputes between the user (article) and other users in articles of varying degrees of controversy. On the constructed social network, WikiNetViz can perform clustering so as to visualize the dynamics of disputes at the user group level. [SCPK07] proposes a model for identifying patterns of conflicts in Wikipedia articles, that relies on users' editing history and the relationships between user edits, especially revisions that void previous edits, reverts. Based on this model, they introduce Revert Graph, a tool that visualizes the overall conflict patterns between groups of users. It enables visual analysis of opinion groups and rapid interactive exploration of those relationships via detailed drilldowns. Motivated by the fact that social transparency and the attribution of ideas and facts to individual researchers is a crucial part of scientific progress, WikiDashboard [SCKP08] aggregates and surfaces "under the hood" information in Wikipedia.

We base our event detection methods on observing the behavior of the community that supports Wikipedia. We investigate how it is mobilized in the presence of events, and leverage the observed patterns to detect event-related information and dynamic relationships between entities. To this end we also leverage the Edit History as a repository for all the interactions between contributors and articles.

Wikipedia and Events Detection

In the earliest work that proposes to exploit the link between Wikipedia and news events, [Lih04], the authors notice that after being exposed through press citation, an article gets a lot of traffic and leads users to improve its quality.

The work of Ferron and Massa [FM11c, FM11b, FM11a, FM12, Fer12, FM13] provides support to the claim that by leveraging Wikipedia's Edit History and talk pages scholars can unobtrusively observe the various psychological processes are represented in the immediate aftermath of an upheaval, and how they may vary in time. Thus, the study of how collective memories of traumatic events are formed in Wikipedia through debates and discussions, and finally represented in shared narratives can provide a high-level perspective on the process of remembrance. [Pen09] proposes to interpret the web-based encyclopedia Wikipedia as a global memory place where memorable elements are negotiated. The complex processes of discussion and article creation are viewed as a model of the discursive fabrication of memory, therefore they can be can be interpreted as the transition, the "floating gap" between communicative and collective frames of memory. Following this interpretation, [FM11c] lays the foundations for the empirical study of collective memories about traumatic events in Wikipedia. By considering the final article as the representation of the crystallized collective memories, which are socially built through direct edits to the article and discussions in the associated talk page by Wikipedia users, it is argued that Wikipedia is a perfect playground for the study of memory building activities, allowing the empirical study on a large scale of collective memory processes. As a first step, the authors show that during anniversaries, edit activity increases signifi-

cantly for both articles and talk pages related to traumatic events. Focusing mainly on the Egyptian revolution, in [FM11b], the same authors provide evidence of the intense edit activity occurred during the uprisings on the related Wikipedia pages. The process in which a lot of people provided their contribution on the content pages and discussed improvements and disagreements on the associated talk pages as the traumatic events unfolded, is interpreted this as a process of collective memory building. By providing a qualitative analysis, the authors argue that on Wikipedia this process can be studied empirically and quantitatively in real time. Building up on the previous work, in [FM11a], extend their interpretation of the patterns of collaboration in Wikipedia as collective memory building, a continuous, active process of sense-making and negotiation between past and present. The authors provide further evidence that Wikipedia enables researchers to study how people build our cultural representations of the past, by leveraging the Edit History and talk pages to empirically and automatically analyze this phenomenon in real-time and on a large scale. Continuing on the same line of thought, in [FM12], by interpreting Wikipedia as a collective memory place, the authors compare articles about natural and humanmade disasters employing automated natural language techniques, in order to highlight the different psychological processes underlying users' sensemaking activities. In [FM13] it is shown that the relative amount of edits during anniversaries can significantly distinguish between pages related to traumatic events and other pages. Moreover, the editing activity of articles about traumatic events can be interpreted as a sign of active participation in remembrance, by composing a unique representation through sharing and compiling of various pieces of information.

[WJA14] provides an overview of the characteristics and related work which support Wikipedia for time aware information retrieval research. The study emphasizes Wikipedia's temporal characteristics that make it a promising source for event detection: freshness, timeliness, a high topic coverage, and quality and correctness. Moreover it points out the temporal signals that can be noticed in the Wikipedia Edit History: temporal expressions, temporal link graph, the page edit and view streams as well as the Current Events Portal. The characteristics that make Wikipedia a valuable source for time-aware research are: the availability of both historical and real-time data, its high topical coverage, its multilinguality, the speed of reaction of their contributors when new events happen, and the multiple levels of structure that constantly evolve. Case studies [KGC11, KGC12] have shown that an increasing level of activity is often triggered by users reporting on ongoing events, as they happen or very soon after. Moreover, in [OPM+12], based on statistics on page views the authors argue that Wikipedia lags between Twitter by about two hours. However, using edit statistics, in [SVHS13] the lag is estimated to be within 30 minutes.

Keegan et al. [KGC11, KGC12] have argued that breaking news articles on Wikipedia offer a compelling case to examine how online communities balance the competing interests to support openness, flexibility, and autonomy against institutional needs for structure, norms, and socialization over very different time scales. In [KGC11]

the authors analyze the patterns of activity on Wikipedia following the 2011 Tōhoku earthquake and tsunami to uncover the dynamics of editor attention and participation, the practices employed to collaborate, and the resulting emerging coauthorship structures between editors and articles. By analysing Wikipedia's coverage of a breaking news event interesting parallels between the goals of Wikipedia and traditional journalism can be observed. Using the revision histories of Wikipedia articles about commercial airline disasters, in [KGC12] the authors construct "article trajectories" that capture the structure and temporal dynamics emerging from the relationships among editors modifying other editors' contributions within an article. Thus, it could be observed that article revision patterns immediately following unexpected, catastrophic incidents differ from the revision patterns of similar articles about historical events. Therefore, breaking news and events trigger patterns of activity that can be observed, and these patterns are different from those exhibited for other types of edit activities.

There have been many approaches for the extraction of events and temporal facts from Wikipedia proposed in the literature. The majority of them focus only on the current version of the Wikipedia articles and ignore their evolution, recorded in the Edit History.

[BFGM07] presents an approach to mining information relating people, places, organizations and events extracted from Wikipedia and linking them on a time scale. The approach consists of two phases: identifying relevant pages containing people, places or organizations and generating timelines by linking named entities and extracting events and their time frame. The result is a collection of pages belonging to the predefined categories and a dynamic graph showing named entities (people, places and organizations) and relations between them. This method is nevertheless based only on the current version of Wikipedia and cannot capture fast dynamics. Starting from the deep parsing of a set of English Wikipedia articles, [ARM+11] produced a semantic annotation compliant with the Knowledge Annotation Format (KAF), that are then structured as a set of RDF triples linked to both DBpedia and WordNet. Also investigating the relationships between entities, [TOY+11] proposes a method to evaluate the significance of historical entities (how they affected other historical entities). The impact of a historical entity varies according to time and location. Assuming that a Wikipedia link between historical entities represents an impact propagation, an iteration algorithm propagates initial tempo-spatial information through links, so that the tempo-spatial impact scores of all the historical entities can be calculated. HistoryViz [SBF+09] is a web application allowing user to explore events (extracted from the current version Wikipedia) connected with selected persons presented on a timeline and to browse the network consisting of persons described on Wikipedia. Although we follow the same goal in our work, to reveal the dynamic relationships that occur during events, the methods we employ rely on the edit history, and not on the current snapshot.

The crowd that contributes to Wikipedia also recognized the need of exposing

and curating events. Therefore, specific methods are employed in order to preserve and document important happenings, and their outcome in Wikipedia pages can be leveraged as sources for event detection and tracking. As Wikipedia collects historical events of different granularity in lists for centuries, years, months and on a daily basis, that are user-maintained with a high actuality and correctness, [HL12] proposes to extract events from articles the for years, leveraging their availability in different languages and the compromise between number and abstractness of events. Another source of events in Wikipedia is the Current Events Portal. This is analyzed in [TA14] showing that it has reached a stable state in terms of the volume of contributions as well as the size of its crowd, thus becoming an important source of news summaries for the public and the research community. Moreover, the authors introduce the WikiTimes project to provide structured access to the Current Events Portal.

The great success of Wikipedia and algorithmic advances in information extraction have lead to an increased interest in large-scale knowledge bases, to which recently attention was diverted towards adding a time dimension. Notable efforts on automatic ontology construction include DBpedia [ABK+07], KnowItAll [ECD+05, BCS+07], WikiTaxonomy [PN09, PS11], Intelligence in Wikipedia [WWA+08], and YAGO [SKW07, SKW08], and meanwhile there are also commercial services such as *freebase.com*, *trueknowledge.com*, or *wolframalpha.com*. These contain many millions of individual entities, their mappings into semantic classes, and relationships between entities. The two most notable initiatives are YAGO and DBpedia. DBpedia has harvested facts from Wikipedia infoboxes at large scale, and also interlinks its entities to other sources in the Linked Data Cloud, while YAGO has paid attention to inferring class memberships from Wikipedia category names, and has integrated this information with the taxonomic backbone of WordNet. While YAGO reuses WordNet and enriches it with the leaf categories from Wikipedia, the DBpedia project has manually developed its own taxonomy. Building up on YAGO, the work on Timely-YAGO [WZQ+10] focused on extracting relevant timepoints and intervals from semistructured data in Wikipedia: dates in category names, lists, tables, and infoboxes. It is the first attempt at automatically constructing ontologies with specific consideration of temporal facts and does not aim at the exhaustive anchoring of an ontology in time and space. It uses regular expressions to extract temporal facts from Wikipedia infoboxes and category names, with a focus on the football domain. YAGO2 [HSBW12], is an extension of the YAGO knowledge base, in which entities, facts, and events are anchored in both time and space. YAGO2 is built automatically from Wikipedia, GeoNames, and WordNet. We are interested in the time dimension of YAGO2, that is considering temporal information for both entities and facts. Entities are assigned a time span to denote their existence in time, while facts are assigned a time point if they are instantaneous events, or a time span if they have an extended duration with known begin and end, based on flexible extraction rules. Both YAGO2 and T-YAGO have extracted temporal facts from Wikipedia, with focus on infobox attributes [HSBW12], on one hand, and a broader range of semistructured elements

but with thematic customization and restriction to the football domain, on the other hand [WZQ+10]. Building up on them, [KW12b] proposes a complete information extraction framework that harvests temporal facts and events from semi-structured data and free text of Wikipedia articles to create a temporal ontology. The temporal knowledge base is built using a information extraction method which harvests temporal facts and events from Wikipedia infoboxes, categories, lists, and article titles. Temporal facts are extracted from structured and unstructured text with the help of patterns for given relations. While this line of work focuses on adding a temporal dimension to knowledge bases based on Wikipedia, either by extracting temporal facts or events, it is limited to the current version of Wikipedia and does not take into consideration the rich dynamics encoded in the Edit History. We on the contrary leverage just the patterns observed in the edit behavior in order to detect events, without having in mind the higher goal of creating or extending a knowledge base.

Event detection can also be done by leveraging the page view history of Wikipedia articles, and a few methods have been proposed in the literature. [CN10] presents a recommender system for new and popular articles, based on favorited Wikipedia articles and page views. Concepts with increased popularity for a given time period are identified by analyzing the trends in page view statistics. The Wikipedia link graph and the Spreading Activation algorithm are leveraged in order to identify topics related to a context. In [OPM+12], the authors explore to which extent event detection, in particular first story detection, based on Twitter can be improved using Wikipedia (when viewed as a stream of page views). By parallely tracking events in Twitter and Wikipedia pages that exhibit abnormally large spikes in page views the authors compare the resultant tweets and Wikipedia pages over textual and time dimensions and identify the common information that is leveraged for filtering spurious events. Moreover they suggest that events within Wikipedia tend to lag for about 2 hours behind Twitter. Also leveraging the page view history, [PMdR] introduces an interface that captures insights into the zeitgeist of Wikipedia users by clustering and comparing concepts based on the time series of the number of views of their Wikipedia pages. Three examples of time-aware information access scenarios in which such a need arises naturally are presented: data mining expert, brand analyst and Wikipedia moderator. In [AVDCB11] the authors show how the page view statistics, along with other features like article text and inter-page hyperlinks, can be used to identify and explain popular trends in Wikipedia. A pipeline for article selection, clustering, and textualization is used in order to identify and describe significant current events as according to Wikipedia content, and metadata. In contrast to this line of work, we are employing the Edit History for event detection. Nevertheless, we similarly leverage spikes in the activity as signals for the occurrence of events.

The Edit History is a rich source of information that can be leveraged for event detection. The methods proposed in this thesis make use of it in order to extract event-related information and explore the dynamic relationships between entities that occur because of events. Observing that news events trigger edit volume variations

in Wikipedia pages, [BL07] investigates if pages that show parallel behaviour in their edit variance are similar or have a co-revision link connecting them. A drawback of the presented approach is that the covariance of two pages is defined based on the entire lifetime of two pages and the number of weekly edits. Based on the edit correlation and revision correlation, the authors try to find similar pages. In our work we evaluate how the similar edit behavior in the same limited time period points to involvement in a common event. Wikipedia Live Monitor [SVHS13] is an application that monitors article edits on different language versions of Wikipedia as they happen in realtime, in order to detect concurrent edit spikes that may be the source of breaking news events. When a concurrent edit spike has been detected, cross-language full-text searches on social networks are used as plausibility checks to filter out false-positive alerts. [SVHS13] also counters the claim of [OPM+12] that Wikipedia lags about two hours behind Twitter, providing an "educated guesstimation" that the lag time for breaking news is of about 30 minutes, and for global breaking news in the range of five minutes and less.

Toolkits

Various tools have been developed for working with Wikipedia. A recently introduced toolkit is Wikipedia Miner [MW13], an opensource software system that allows researchers and developers to integrate Wikipedia's rich semantics into their own applications. The toolkit creates databases that contain summarized versions of Wikipedia's content and structure, and includes a Java API to provide access to them. Wikipedia's articles, categories and redirects are represented as classes, and can be efficiently searched, browsed, and iterated over. Advanced features include parallelized processing of Wikipedia dumps, machine-learned semantic relatedness measures and annotation features, and XML-based web services. If one requires a thesaurus or ontology derived from Wikipedia rather than direct access to the original structure, there are several options, such as Dbpedia [ABK+07], Freebase [BCT07], YAGO [SKW08], BabelNet [NP12], or Menta [dMW10].

In our work we make use of the Java Wikipedia Library(JWPL) [ZMG08], in particular of the Wikipedia Revision Toolkit [FZG11a], which is an open-source toolkit that allows to reconstruct past states of Wikipedia, and to efficiently access the Edit History of Wikipedia articles. By using a dedicated storage format, this toolkit massively decreases the data volume to less than 2% of the original size, and at the same time provides an easy-to-use interface to access the revision data. Basically, instead of storing each revision of an article, only the differences between consecutive revisions are stored, in order to save space.

3

Extraction and Summarization of Event-Related Wikipedia Updates

Wikipedia articles and associated edits constitute a potentially interesting data source to mine for obtaining knowledge about real-world events. In this chapter, we conduct a study on this information with several complementary goals. On the one hand, we study the viability of using the Edit History of Wikipedia for extracting event-related updates. This has direct applications to building annotated timelines and news tickers for specific entities featured in Wikipedia articles such as persons and countries. On the other hand, we perform an in-depth analysis of event-related updates in Wikipedia, including qualitative and quantitative studies for sets of samples gathered using different filtering mechanisms.

We first conduct an in-depth analysis of event-related updates in Wikipedia by examining different indicators for events including language, meta annotations, and update bursts. We then study how these indicators can be employed for automatically detecting event-related updates. In order to extract event-related information from Wikipedia edits for a given entity and its corresponding article, we first identify event-related updates and then, in a second step we cluster these updates in order to map the updates to their corresponding events and to generate summaries (cf. Figure 3.1). Moreover, we present *Wikipedia Edit Reporter*, an application based on the proposed methods.

In order to identify event-related updates we employ different filters and extraction methods. First, we apply burst detection because events of interest tend to trigger peaks of attention for affected entities. Date detection helps to identify event-related updates that contain dates in the proximity of the update creation time. Finally, we build classification models based on the textual content of the updates as well as meta annotations. To summarize event-related information, we perform clustering of edits by exploiting different types of information such as update time, textual similarity, and the position of edits within an article.

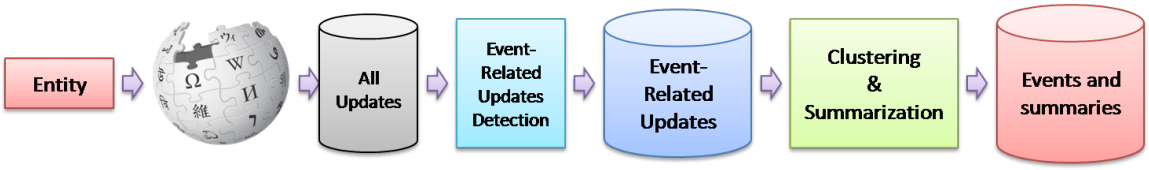


Figure 3.1 Pipeline for identifying and presenting the events related to an entity.

3.1 Event Extraction Methods

An *update* in Wikipedia represents the modifications present in one revision when compared to the previous revision of an article. It is accompanied by its creation time (timestamp), its author, and, possibly, comments provided by the updater. For a given update, we further consider the blocks of text added and removed, the title of the section where the modification occurred, and the relative and absolute positions of the blocks in their sections and in the article.

In order to extract event-related information from Wikipedia edits for a given entity and its corresponding article, we first identify event-related updates; in a second step we cluster these updates in order to map the updates to their corresponding events and to generate summaries. The pipeline for this process is depicted in Figure 3.1. In the following sections we describe the methods we employ for event-related update detection and summarization.

3.1.1 Detection of Event-Related Updates

For detecting event-related updates we make use of a combination of filters and classifiers based on burst detection, temporal information, and textual content.

Burst Detection Filter: Bursts of updates (peaks in the update activity) in a Wikipedia article are indicators for periods with an increased level of attention from the community of contributors. As we will discover later in our analysis in Section 3.2, bursts often co-occur with real-life events, making burst detection a promising filter for gathering event-related updates. In order to detect bursts, we apply a simplified version of the burst detection algorithm presented in [ZS03] on the temporal development of the update frequency of an article. The algorithm employs a sliding time window for which the number of updates is counted. The corresponding time intervals for which the update rate exceeds a certain threshold are considered *bursty*; our burst detection filter extracts the updates within those bursty periods. The parameters of the algorithm are ω - the size of the sliding window (e.g., day, week, or month), and θ - a threshold for the number of standard deviations above the average update number over the whole lifetime of the article for a time interval to be considered as bursty. All the updates contained in the identified bursty windows will pass the filter to go to the next step in the pipeline.

Date Extraction Filter: This filter makes use of the following heuristic: If the textual content of the update contains a date which is in close temporal proximity to the timestamp of the update, then this is an indicator that the update might be connected to an event. More specifically, our filter identifies temporal expressions in updates matching the format recommended by Wikipedia[1], and checks if these expressions fall into the interval within one month before or after the update was done.

Text Classification: Language and terms used in the update text can serve as an indicator whether an update is related to an event. For instance, we observed that terms like *death*, *announce*, and *outburst* are typical for event-related updates. In addition, Wikipedia updates are often accompanied with meta annotations such as "{current}" (explicitly marking current events) or "rvv" in comments (indicating vandalism rather than events) which can provide additional clues on the event-relatedness of updates. In order to exploit that type of information we trained Support Vectors Machine classifiers [CV95] on manually labeled samples to distinguish between "event-related" and "not event-related" updates. We tested different bag-of-words based feature vector representations of updates, which will be described in more detail in Section 3.3.

3.1.2 Clustering and Summarization of Event-related Updates

The stream of event-related updates determined in the previous step serves as a starting point for identifying the events themselves and creating a meaningful summarization. In order to present event-related information in a understandable way, instead of using the detected event-related updates for summarization, we use the sentences that were modified by them. To this end, we start by identifying the sentences where the event-related updates were done, and assign to them a *Weight*, corresponding to the number of times they were updated, and a list of positions at which the sentences appeared within the Wikipedia articles.

Temporal Clustering: As already observed in Section 3.1.1 events are signaled in Wikipedia by a burst of updates. Therefore, in order to identify the distinct events, we first resort to a temporal clustering by identifying the bursts among the event-related updates. Each burst of event-related updates corresponds to a distinct event.

Text-Based Clustering: Within a burst of updates, in order to eliminate the duplicate sentences and group together the sentences that treat the same topic we employ an incremental clustering based on the Jaccard similarity as a distance measure. Each *Sentence cluster* is characterized by the *Aggregated weight* of member sentences, and represented by the longest member sentence, that serves as a candi-

[1] http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Dates_and_numbers

date for summarization.

Position-Based Clustering: Assuming that sentences treating the same topic are located in spatial proximity of each other on the article page, by investigating the positions of all sentences modified in a burst we can identify *Position clusters*. A cluster of positions is a contiguous succession of positions with no more than 10 positions gap in between. Each sentence cluster belongs to the position cluster that has the maximum overlap of positions with member sentences.

Summarizing Detected Events: Each identified event, corresponding to a burst of updates is summarized using a ranked list of sentences. We rank the position clusters by how many sentence clusters are assigned to them, ignoring the position clusters that are not well represented and we rank the sentence clusters by the *aggregated weight* of their member sentences. The proposed summarization for an individual event consists of displaying for each of the top-N identified position clusters, the representative sentences for the top-M clusters of sentences.

3.1.3 Datasets

We downloaded the dump of the whole Wikipedia history (version from 30 January 2010). The history dump contains more than 300 million updates with the size of approximately 5.8 TB covering the time period between 21 January 2001 and 30 January 2010. We discarded updates made by *anonymous* users, resulting in a dataset containing 237 million updates belonging to 19 million articles. We studied our proposed method for extracting event-related information using different datasets created by randomly selecting Wikipedia updates for: 1) articles from all categories, and 2) only those belonging to the *people* category. Note that, we discarded all the articles that had less than 1,000 updates.

By considering articles from all categories, we can investigate the domains on which our proposed methods can be applied without any limitation on some particular article type. In this case, we sampled updates in three ways:

- **ALL-Random** was collected by randomly sampling from all available updates in our history dump collection.
- **ALL-Burst** was collected taking into account the time dimension by sampling updates coming from bursts, where bursts were identified by using the detection algorithm described in Section 3.1.1 with the empirically chosen parameters $\omega = 2$ days and $\theta = 4$.
- **ALL-Date** was gathered using a constraint in which article updates contain at least one *date mention* in the proximity of their timestamps. More precisely, we checked whether the month and year of timestamps occurred inside the text added, removed or inside the comments. This dataset was also selected from burst periods determined using $\omega = 2$ days and a higher $\theta = 32$ in order to

filter just the updates done in highly salient bursts and to increase the chances of finding event-related updates.

In addition to the selection methods described above, we investigated updates of Wikipedia articles from the category *people* in particular because the updating of personal information is highly relevant to some events, e.g., professional achievement, changing of civil status, or health issues. We randomly selected 185 Wikipedia articles, whose categories start with "peopl" and contain at least a burst of updates. In detail, we sampled updates in three ways:

- **PPL-Burst** was created by randomly selecting 10 updates for each article coming from the identified bursts using $\omega = 2$ days and $\theta = 12$. The parameters of the burst detection algorithm were chosen in order to offer a reasonable number of candidates to sample from.
- **PPL-Date** was collected by randomly choosing 10 updates for each article with dates in the vicinity of their timestamps, i.e., in the window of one month before/after timestamps. Date mentions were identified by looking for date mentions in the standard formats provided by Wikipedia. Note that, we filtered out date mentions found in an administrative context because they might not be related to events.
- **PPL-Random** was created by randomly selecting 10 updates for each article without considering bursts or containing date mentions close to their creation timestamps.

Our last dataset, denoted **DETAIL**, was created by selecting four particular entities: Jerry Fallwell, Donald Rumsfeld, Alexandr Solzhenitsyn and Kosovo. Each of those entities is associated to one or more important events, and we aimed at performing a detailed analysis of bursts. For each article, we used all updates from bursts identified using the narrower parameter choice, $\omega = 2$ days and $\theta = 32$, in order to perform further investigation of update dynamics.

3.2 Data Analysis

In this section, we perform an in-depth analysis of event-related updates in Wikipedia gathered using the different filtering mechanisms as explained in the previous section.

3.2.1 Data Labeling

There exists no ground truth dataset for evaluating the task of event extraction from Wikipedia updates. In order to identify which of the updates are related to events

we therefore manually labeled the updates in the datasets described in the previous section. More precisely, for each article update we provided a human assessor with the *differences* (i.e., text added or removed) between the revision before and after the update using Wikipedia's *diff* tool [2]. In addition, we provided the *comment* made by the editor of an update as additional context. The human assessor was asked to assign one of the following labels to each update: "event-related" or "not event-related". The updates on which the assessor was unsure about, were discarded in the experiments and analysis. Vandalizing updates were regarded as *not event-related*. For the event-related updates, we also determined whether they were *controversial* or not. An update was considered as controversial if it: 1) contained a point of view, 2) was repeatedly added and removed, and 3) exhibited a dispute between the contributors. These annotations help to understand the effect of controversy in the process of updating an article in the case of an event, and show how many of the event related updates are likely to be disputed. In order to gain further insight into the types of edits that occur during bursty periods, we performed a detailed investigation by categorizing them into the following classes:

- *fact:* the update modified the facts presented in the article,
- *link:* the update was made by adding/removing links within or outside Wikipedia,
- *markup:* the update only changed the cosmetic appearance or Wikipedia markup,
- *vandalism:* the update consisted of vandalism or the reaction to vandalism,
- *spelling:* the update consisted of edits done to the punctuation, spelling or formulation of text without modification of the underlying facts
- *category:* the update consisted of changes done to the the category of a Wikipedia article.

Finally, there were approximately 10,000 article updates labeled and the dataset is publicly available for download[3].

3.2.2 Data Statistics

Table 3.1 shows statistics of our datasets including the total number of labeled updates, the number of *event-related* updates (number of *controversial* updates in parentheses), and the number of *non event-related* updates (number of *vandalism* updates in parentheses). We observe that filtering by bursts increases the number of event-related updates found. The percentage of event-related updates for **ALL-Burst** increases up to 10% compared to just 1% for **ALL-Random**. The burst detection

[2] http://en.wikipedia.org/w/index.php?diff=prev&oldid=[REVISION_NO]

[3] http://www.l3s.de/wiki-events/wiki-dataset.zip.

Table 3.1 Statistics of the datasets in terms of total number of updates, and the type of updates.

Dataset	Updates	Event-related (Controversial)	Unrelated (Vandalism)
ALL-Random	961	13(0)	948(63)
ALL-Burst	1331	133(21)	1198(141)
ALL-Date	1626	1037(256)	589(51)
total	*3918*	*1183(277)*	*2735(255)*
PPL-Random	1850	62	1788(329)
PPL-Burst	1850	199	1651(159)
PPL-Date	1448	604	844(310)
total	*5148*	*865*	*4283(798)*
DETAIL	1614	568(280)	1046(108)

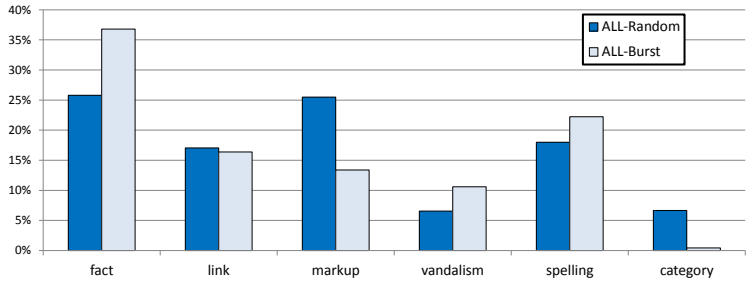


Figure 3.2 Classes of updates.

increases the number of event-related updates from 3% in **PPL-Random** to 11% in **PPL-Burst**, amplified to 41% in **PPL-Date**. We further observe a substantial increase in the number of *event-related* updates when filtering by date mentions. For the **ALL-Date** dataset, 66% of the updates are related to events, and 30% of those are controversial. More event-related updates took place during bursty periods showing that burst detection helps increasing the percentage of event-related updates while reducing the overall number of updates to choose from. This effect can be further amplified by using date filtering. The number of vandalism updates is steady across our samples with a slight increase in the case of the **ALL-Date** and **PPL-Date** samples.

Figure 3.2 illustrates the percentage of updates labeled into different classes for **ALL-Random** and **ALL-Burst**. We can observe differences between updates made in general and updates made during bursty periods. The samples taken from the detected bursts contain substantially more updates related to facts rather than changing the cosmetic appearance and style of the articles.

Table 3.2 Statistics for the Details dataset.

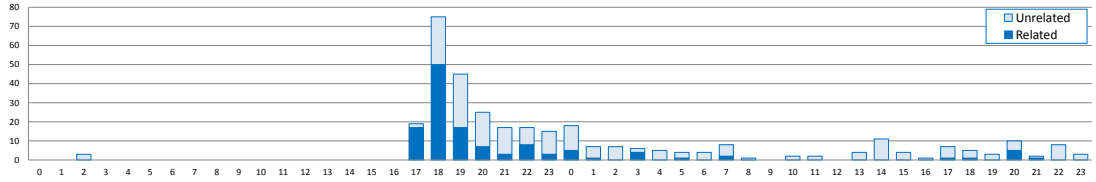
Dataset	Updates	Event-related (Controversial)	Unrelated (Vandalism)
Jerry Fallwell	454	127(37)	327(35)
Donald Rumsfeld	338	126(59)	212(41)
Alexandr Solzhenitsyn	130	36(1)	94(5)
Kosovo	692	279(183)	413(24)

3.2.3 Investigating Bursts of Updates

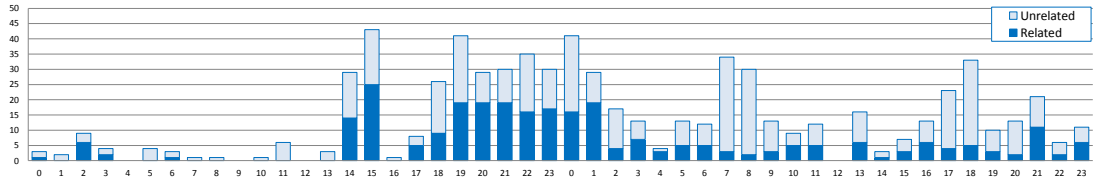
We investigated the updates made on the four articles in the **DETAIL** dataset in order to better understand the process of event-triggered updating. Statistics on the componence of this dataset are presented in Table 3.2. Figure 3.3 shows the distribution over time of the number of updates for the Wikipedia articles on Donald Rumsfeld, Kosovo, Jerry Falwell and Alexander Solzhenitsyn. For every hour of the day since the beginning of the burst, we plot the number of updates composed of the event-related and not event-related updates. We observe that *not* all of the updates done during a burst period are related to an event. After a burst, the updates are no longer related to the events; instead, the attention is rather directed towards making the article more accurate, giving raise to correction of unrelated facts, punctuation and cosmetic changes. For the resignation of Donald Rumsfeld, we notice that the burst of updates contains a small number of peaks, which are bigger at the beginning of the event and then become smaller as the overall number of updates and the number of event-related updates decrease towards the end of the burst. This might be a characteristic of the type of event or entity. If the event is not controversial, or no other information becomes available, the interest in editing the article drops. In contrast, if the entity or the event is controversial or the event develops over a longer period of time, as in the case of Kosovo's independence declaration, the interest decreases much slower.

3.2.4 Discriminative Term Analysis

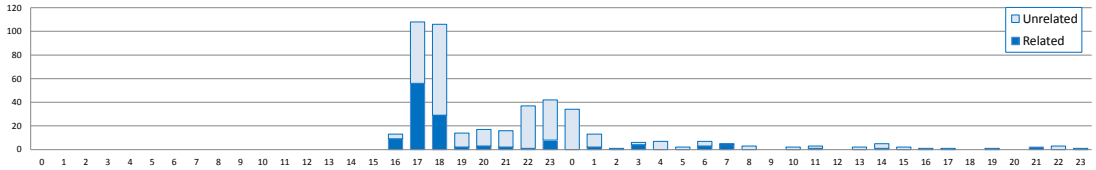
In order to assess the feasibility of building a term-based classifier, we studied the differences between the terms used in event-related updates and non event-related updates by conducting a discriminative term analysis . For computing ranked lists of stemmed terms from the set of event-related updates, and the updates unrelated to events, we used the information-theoretic Mutual Information (MI) measure [MS99]. It can be interpreted as a measure of how much the joint distribution of features (terms in our case) deviates from a hypothetical distribution in which features and categories ("event-related" and "not event-related") are independent of each other. Table 3.3 shows the top-20 stemmed terms computed from the datasets containing a sufficient



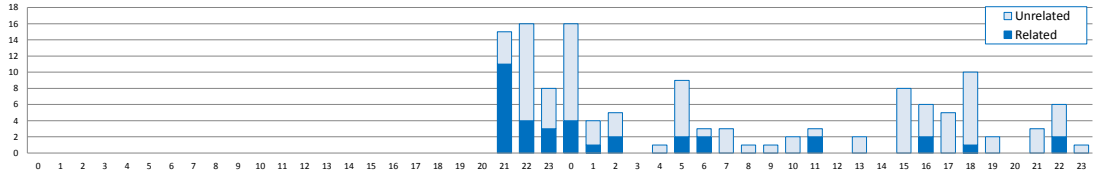
(a) **Donald Rumsfeld** and the corresponding event of *resignation*.



(b) **Kosovo** and the corresponding event of *independence declaration*.



(c) **Jerry Falwell** and the corresponding event *death*.



(d) **Alexander Solzhenitsyn** and the corresponding event *death*.

Figure 3.3 Distribution over time (in hours) of updates for selected Wikipedia articles: Donald Rumsfeld, Kosovo, Jerry Falwell and Alexander Solzhenitsyn.

Table 3.3 Top (stemmed) terms ranked by MI values for two types of updates.

Dataset	Event-related Terms	Not Event-related Terms
PPL-Date	2006 second state schedul date add championship announc time releas presid report current year publish contract news titl sport web	2007 2004 sysop delet excess 18 use 15 juli protect expir march level wp:vandal decemb expiri autoconfirmed:mov 22 edit utc
ALL-Date	reaction stori 2009 2006 2007 state bhutto 12 report die presidenti wilson decemb obama www.cnn.com 08 news death outburst septemb	squar common tavistock street use wikifi pancra king bma network destroy life page fix name woburn power edgwar terrorist russel april

number of event-related updates. For all of the updates we considered words added and removed, as well as words from comments and meta annotations denoting the type of the update. We observe that time-related terms (*date, time, current*), sports-event related terms (*championship, sport, schedul*), news-related terms (*news, announc, publish, releas, stori, report*) or status change terms (*die, death, outburst*) characterize the event-related updates as opposed to Wikipedia administrative terms (*sysop, delet, wikifi, page*) or general terms (*common, street, king, power*) that characterize updates that are unrelated to events. These differences provide a good indication that the terms are good features for representing the updates in a classification task.

3.2.5 Further Insights into the Wikipedia Update Process

There are different causes for a burst, an event being just one particular case. We present some of the patterns for the update bursts causes observed in our datasets. An article can be the featured article for the day, triggering a lot of updates that will contribute to an improvement of its quality. An anniversary of an entity, and the celebration will cause extra interest. Articles sometimes go through a major expansion or restructuring. Vandalism also causes update bursts. Periodical events such as Halloween, cause interest that trigger bursts, without having anything to to with the event itself.

In the case of event triggered bursts we observed further patterns. In all the cases that involve a change of status: from alive to dead, from in office to resigned, there will be a lot of activity, between people inserting information without having good references, and it being removed by other users requiring authoritative references. Sometimes comments are added to prevent people altering the information until a safe source is found. Updates that change the tense of verbs from present to past can also be noticed. Usually deaths are signaled by the template announcing a recent death, removal from the living people category, addition to deaths in current year category. Some other past events, not related to the event taking place may resurface in the updates. Time related adverbs and dates are a clear sign of event-related

updates.

There are cases when two entities involved in the same event simultaneously get a burst of updates, one observed example consisting of the entities "Steve Irwin" and "Stingray". The TV star Steve Irwin died while filming a documentary, because a stingray barb pierced his chest. Besides the updates to the article dedicated to Steve, the Stingray article was also updated with this information, which got repeatedly removed because it was not regarded as a good encyclopedic practice. This supports the investigation we perform in Chapter 4.

One special case of event-related updates to an entity is the case of movies and books, which after their release will have their summaries posted on Wikipedia. We may consider this as an event about which information is gradually becoming available with the passing of time. Unfortunately, other users were removing these updates, regarding them as spoilers. This behavior can be regarded as active censorship.

Sometimes Wikipedia is used as a source of information regarding recent events as is the case of the "7th July 2005 London bombings". During the event and in the near time, the article had no photos, and a request not to add any in order to minimize the download time could be read on the articles' page. We have also observed users that are complaining that sometimes Wikipedia behaves as a news portal, contrary to their expectations, and thus explicitly request other users to keep their contributions between the lines of an encyclopedia, in their updates making comments like "This is Wikipedia, not fox News" or "Removed 'Breaking News' entry: This is an Encyclopedia, not a News blog".

3.3 Evaluation of Event-Related Information Extraction

In this section, we investigate closer the components of the pipeline described in Section 3.1, by evaluating methods for event-based classification as the final step in the detection of event-related updates and presenting some examples of extracted and summarized events.

3.3.1 Event Classification

For text-based classification of updates into categories "event-related" and "not event-related" we used the LIBSVM [CL11] implementation of linear support vector machines (SVMs) with the default parameters.

We conducted our evaluation on **ALL-Burst**, **ALL-Date**, **PPL-Burst**, and **PPL-Date** as these datasets contain a sufficient number of event-related updates for experiments (cf. Section 3.2). We experiment with different feature representations of the updates. If some of these feature representations generate empty documents,

they are excluded from the experiments. To avoid an imbalance towards one category or the other, we randomly chose a number of instances from the larger category equal to the number of instances contained in the smaller category. For testing the classification performance on the thus generated balanced datasets we used 5-fold cross-validation. We repeated this procedure 100 times and averaged over the results.

The quality measures we use are precision, recall as well as the break-even points (BEPs) for precision-recall curves (i.e. precision/recall at the point where precision equals recall, which is also equal to the F1 measure, the harmonic mean of precision and recall in that case). We also computed the area under the ROC curve values (AUC) [Faw06]. ROC (Receiver Operating Characteristics) curves depict the true positive with respect to the false positive rate of classifiers.

We compared the following update representations for constructing bag-of-word based tf*idf feature vectors (using stemming and stop word elimination for each of the options):

- *wordsAdd* - terms added in an update
- *wordsRmv* - terms removed in an update
- *All* - terms added in an update, terms removed, and terms from comments
- *P.text* - terms from text added and removed treating added and removed terms as different dimensions in the feature vector
- *P.all* - terms added in an update, terms removed, and terms from comments treating added, removed, and comment terms as different dimensions in the feature vector
- *P.T.all* - *P.all* with the titles of the updated sections as additional context

Table 3.4 shows the results of our experiments. We achieve the best performance for the feature representation using a combination of terms added in an update, terms removed, and terms from comments (*All*), with an AUC value of 0.87 and a BEP value of 0.79.

We notice that using just the words added or the words removed alone is performing worse than using all the data available. Taking into consideration the provenance of the words or the section title does not provide a significant increase in performance. The addition of the comment gives just a small boost in the setting where the provenance is taken into account. Using a simple model that puts all the words in the update together, provides a good balance between classification performance and model complexity.

Table 3.4 Classification performance using different textual representations.

Features	ALL-Burst				ALL-Date			
	AUC	BEP	P	R	AUC	BEP	P	R
wordsAdd	.75	.69	.77	.53	.76	.70	.72	.58
wordsRmv	.78	.72	.82	.53	.70	.66	.69	.56
All	.80	.73	.80	.54	.80	.74	.78	.61
P.text	.75	.68	.78	.51	.75	.69	.72	.58
P.all	.76	.69	.77	.51	.77	.71	.74	.58
P.T.all	.73	.67	.74	.47	.72	.68	.71	.52
Features	PPL-Burst				PPL-Date			
	AUC	BEP	P	R	AUC	BEP	P	R
wordsAdd	.75	.69	.77	.53	.77	.71	.77	.57
wordsRmv	.78	.72	.82	.53	.73	.67	.70	.62
All	.80	.73	.80	.54	.87	.79	.81	.78
P.text	.75	.68	.78	.51	.77	.71	.76	.60
P.all	.76	.69	.77	.51	.86	.79	.78	.82
P.T.all	.73	.67	.74	.47	.81	.74	.70	.89

3.3.2 Clustering and Summarization of Event-Related Updates

Table 3.5 shows some example outputs of the clustering and summarization step described in Section 3.1.2. For each event we show its date and the top-2 sentence cluster representatives along with the cluster weight. For Paul Newman the event detected is his death. Most of the edits occurred in the introduction of his Wikipedia entry, where contributors added his death date. The high number of edits is due to the sentence having been added and removed several times until a trusted source confirmed the information. The second sentence provides more details about his death. For Donald Rumsfeld the most frequently edited sentence is the announcement of his planned resignation, and the second most frequently edited one is related to the nomination of a successor and includes a link to the mainstream media. For Charlie Sheen the summarized event that drew the attention of the Wikipedia community is his provocative comment on the 9/11 attacks.

Table 3.5 Examples of extracted and summarized events.

Entity	Event date	Weight	Representative Sentence
Charlie Sheen	12 September 2009	26	Days before the eight anniversary of the 9/11 attacks, Sheen publicly requested a meeting with President Obama to discuss a list of 20 questions he had about the September 11th attacks which he says remain unanswered and is demanding an investigation into the attacks be reopened
Charlie Sheen	12 September 2009	19	On September 8, 2009, Sheen released an open letter to President Barack Obama outlining his concerns and questions relating to a possible new investigation into the WTC attack.
Steve Irwin	Sep 4, 2006	36	He was best known for the television program " The Crocodile Hunter ", an unconventional nature documentary wildlife documentary series broadcasted worldwide and hosted with his wife Terri Irwin Terri ; the program gave him his sobriquet.
Steve Irwin	Sep 4, 2006	20	After he was stung, his crew called for medical help and the Queensland Rescue Helicopter responded. However, Irwin was immediately pronounced dead at the scene.
Paul Newman	27 September 2008	9	"'Paul Leonard Newman"' (January 26, 1925 - September 26, 2008)
Paul Newman	27 September 2008	5	On September 26, 2008, Newman died at his long-time home in Westport, Connecticut, of complications arising from cancer
Donald Rumsfeld	8 November 2008	13	On November 8th, 2006, the GOP announced that Rumsfeld plan to resign from his position as Defense Secretary.
Donald Rumsfeld	8 November 2008	11	President Bush has nominated Robert Gates, former head of the CIA, to replace Rumsfeld http://www.cnn.com/ 2006/POLITICS/11/08/ rumsfeld.ap/index.html

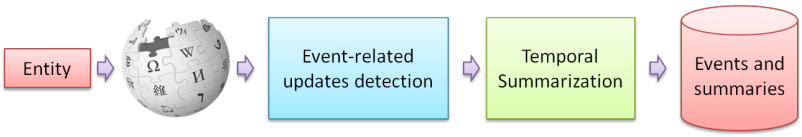


Figure 3.4 Wikipedia Edit Reporter pipeline for temporal summarization of event- related information from the Wikipedia article updates for an entity.

3.4 A System for Temporal Summarization of Event-Related Information from Wikipedia Updates

In this section we propose an application of the previously presented methods for temporal summarization of event-related information extracted from the stream of updates done to Wikipedia.

As an event happens, the Wikipedia community mobilizes itself to update the encyclopedia. Some of the information generated in this time will no longer be available in a future version of the articles of the entities involved in the event. That is the reason why our system, *Wikipedia Edit Reporter*, provides the user with the ability to visualize all the historical information, giving the user a comprehensive view of the event, and not only the socially accepted final interpretation of the event. In order to detect, extract and summarize event-specific information from Wikipedia updates, we use the methods presented in Section 3.1 and Section 3.1.2.

Wikipedia Edit Reporter uses the Edit History of Wikipedia for extracting event-related related information. For extracting event-related information from Wikipedia edits, we first identify event-related updates; then we cluster these updates in order map the updates to their corresponding events and to generate (cf. Figure 3.4). First, we apply burst detection because events of interest tend to trigger peaks of attention for affected entities, and then employ a classifier to detect the event-related updates. To summarize event-related information, we perform clustering of updates by exploiting different types of information such as update time, textual similarity, and the position of edits within an article.

3.4.1 Event Extraction Methods

As a data source we use the dump of the whole Wikipedia history (version from 30 January 2010). To store the revisions in a way that they are easily accessible for processing and information extraction we used the Wikipedia Revision Toolkit [FZG11b].

As previously introduced in Section 3.1, an *update* in Wikipedia represents the modifications present in one revision when compared to the previous revision of an article. The metadata associated to the update consists of its creation time (timestamp), its author, and, possibly, comments provided by the updater.

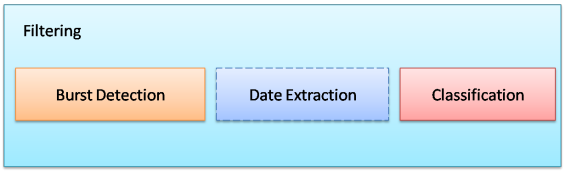


Figure 3.5 Conceptual depiction of the Wikipedia updates filtering component.

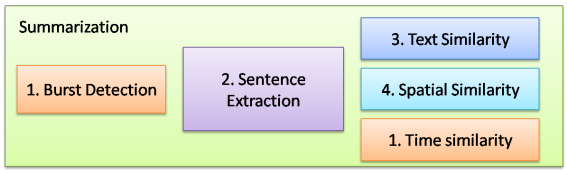


Figure 3.6 Conceptual depiction of the summarization of event-related updates component.

In order to extract event-related information from Wikipedia edits for a given entity and its corresponding article, we employ the methods described in Section 3.1.1. We first identify event-related updates; in a second step we do a temporal summarization of the updates in order to map the updates to their corresponding events and to generate summaries.

The pipeline for this process is depicted in Figure 3.4. The *Event-Related Updates Detection* step is based on a *Burst Detection Filter* followed by a text *Classification* step as depicted in Figure 3.5. We decided to omit the *Date Extraction* component. To represent an update we constructed bag-of-word based tf*idf feature vectors (using stemming and stop word elimination) using the terms added in an update, terms removed, and terms from comments (the *All* feature as described in Section 3.3). As training data we used 10,680 article updates labeled, of which 2,616 as "event-related" and 8,064 as "not event-related". The *Temporal Summarization* of events step, depicted in Figure 3.6, is based on different clustering methods applied in order. We start temporal clustering in order to identify further bursts in the event-related updates. We then *extract the sentences* that are a result of the updates and employ a text-based clustering based on *text similarity*, and then a position-based clustering based on *spatial similarity*. We finalize by presenting the *Events and Summaries* by displaying for each event, each of the top-N position clusters along with the representative sentences for the top-M clusters of sentences.

3.4.2 System Interface

The information that we present about a specific entity consists of a timeline of the updates, annotated with the detected events, a histogram depicting the positions of the edited sentences, and a list of sentences that characterize the event.

In Figure 3.7 we present an example for the timeline of the number of edits per day. Each of the peaks might be a candidate for an event, therefore we employ a classifier to detect the event-related edits, and highlight only those peaks that are actually related to events. The detected events are marked on the timeline with the letter assigned to them, and are accompanied by a tag cloud description.

When clicking on the event from the timeline, the corresponding summary can

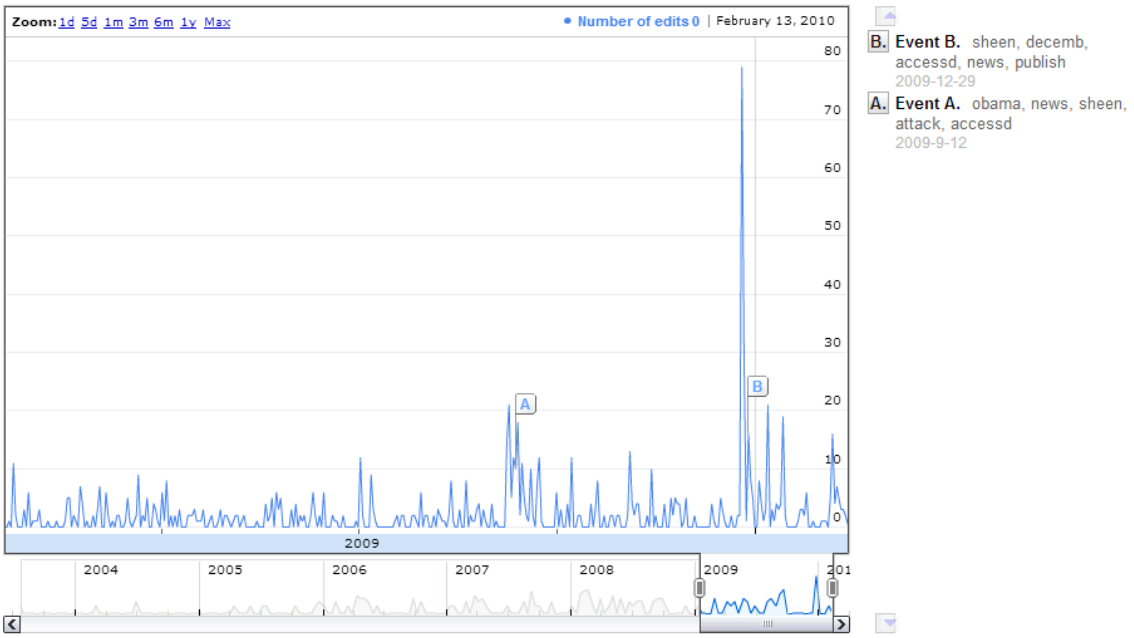


Figure 3.7 Annotated Timeline for Charlie Sheen zoomed in around the detected events.

be displayed for different time granularities: the *Whole Event History* and for each individual *day* that is a part of the event. All updates that were detected as being event related that belong to a common burst, characterize the same event. We cannot offer an intelligible description by using the updates themselves, because they are often just words or parts of sentence. Therefore, we use the sentences that were modified by the updates for summarization.

The histogram called *Positions Histogram* presented in Figure 3.8 represents the positions of all the sentences that were edited that belong to the same event. The histogram is annotated using different colors for the identified positions clusters.

Because there most of the sentences are similar, they are clustered together in *Sentence Clusters*. As it can be seen in Figure 3.9, we rank the clusters based on their *Weight*, and display the top 10 clusters presenting: the weight of the cluster, the representative sentence, the section name where most of the edits were made, and the positions cluster assignment. The color and number of the Positions Cluster assignment match to the Positions Histogram displayed above. When hovering over the positions cluster assignment, the user can see a histogram of all the positions of the sentences that are a part of the cluster. The positions clusters colors are easily identifiable to facilitate the understanding of the positions cluster assignment.

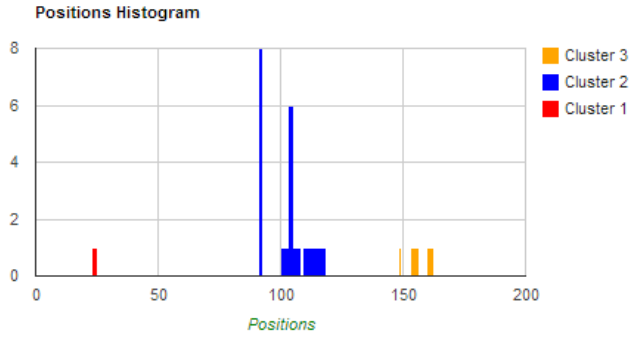


Figure 3.8 Histogram illustrating the positions where the event-related edits occured.

Weight	Sentence Representative	Section Title	Positions Cluster
11	= On March 20, 2006, Sheen stated during an Alex Jones (radio) Alex Jones interview that he questions the official story concerning the September 11 attacks of 2001.	September 11 attacks	2
6	On September 8, 2009, Sheen appealed to US President Barack Obama Obama to set up a new investigation into the attacks.	September 11 attacks	2
5	Presenting his views as a transcript of a fictional encounter with Obama, he characterized the 9/11 commission as "a whitewash" and alleged that the administration of former US President Bush was responsible for the attacks.	September 11 attacks	2
2	{{cite news last=Banerjee first=Subhajit journal=Daily Telegraph title=Charlie Sheen urges Barack Obama to reopen 9/11 investigation in video message date=September 12, 2009 url=http://www.telegraph.co.uk/news/newstopics/celebritynews/6177194/Charlie-Sheen-urges-Barack-Obama-to-reopen-911-investigation-in-video-message.html accessdate=September 13, 2009}}	September 11 attacks	2
1	Charlie Sheen has since become a prominent advocate of the 9/11 Truth movement .	September 11 attacks	2
1	Days before the eight anniversary of the 9/11 attacks, Sheen publicly requested a meeting with President Obama to discuss a list of 20 questions he had about the September 11th attacks which he says remain unanswered and is demanding an investigation into the attacks be reopened.	Charitable and political activities	3
1	Sheen stated that a friend of his died from breast cancer and he wanted to try to help find a cure for the disease.	September 11, 2001	3
1	ref name="CNN Showbiz March"	September 11 attacks	2
1	{{cite journal last1=Keating first1=Joshua last2=Downie first2=James title=The World's Most Persistent Conspiracy Theories journal=Foreign Policy date=September 10, 2009 url=http://www.foreignpolicy.com/articles/2009/09/10/the_worlds_most_popular_conspiracy_theories accessdate=September 13, 2009}}	September 11 attacks	2
1	{{cite news url=http://sports.espn.go.com/espn/page3/story?page=sheen/merron title=How Good Was Charlie Sheen? last=Merron first=Jeff date=2004-02-19 work=Page 3 publisher=ESPN accessdate=2009-03-21}}	September 11 attacks	1

Figure 3.9 Temporal summarization of detected events for a given entity.

3.4.3 System Demonstration

To demonstrate the capabilities of our proposed system we proceed as follows. We show how to use the system in order to visualize and generate temporal summarization for the events about Charlie Sheen's life and career. First, we enter Charlie Sheen as an input query for the system. In this example, the system will display the timeline annotated with the events A and B on date September 12, 2009 and December 29, 2009 respectively as illustrated in Figure 3.7.

By checking the date of the detected Event A we find that it matches the following text in the current Wikipedia article of Charlie Sheen: *On September 8, 2009, he appealed to President Barack Obama to set up a new investigation into the attacks. Presenting his views as a transcript of a fictional encounter with Obama, he was characterized by the press as believing the 9/11 Commission was a whitewash and that the administration of former President George W. Bush may have been responsible for the attacks.* Figure 3.9 represents the temporal summary for Event A provided by the system, and Figure 3.8 depicts the corresponding positions histogram. It can be noticed that the generated summary that the system provides matches the current Wikipedia article. In addition our system presents links and content that are no longer available in the current version because as an encyclopedia,Wikipedia has to keep just the relevant, high quality content. This allows the user to discover details that were removed for the sake of brevity in the current version of Wikipedia such as: *Days before the eight anniversary of the 9/11 attacks, Sheen publicly requested a meeting with President Obama to discuss a list of 20 questions he had about the September 11th attacks which he says remain unanswered and is demanding an investigation into the attacks be reopened.*

In Event B, that is about a domestic dispute, *On December 25, 2009, Sheen was arrested for assaulting his wife, Brooke Mueller in Aspen, Colorado*, most of the information we display is no longer available in Wikipedia: *Law enforcement sources cited by TMZ.com said Mueller initially told 911 dispatchers Sheen had assaulted her, alleging Sheen put a knife to her throat and made threats to kill. Mueller had a blood alcohol level of 0.13 that night (over the legal limit for driving), Sheen's BAC was 0.04 (well under the limit).* None of the links to the articles describing the incident are available in the current Wikipedia version but are discovered by our system. In this case the system uncovers details about an event that were not deemed as worthy of being kept in Wikipedia, but might be of interest to someone studying the entity in detail, that is not satisfied just with what is provided by the last version of the article, or does not want to spend to much effort searching the Web.

An online system and a video tutorial have been made available at http://www.l3s.de/wiki-events. More instructions on all the available summarization tools are available in the *Man* page and a short description of the processes that take place in the background and the tools used can be found in the *Behind the Scenes* page of the online system.

3.5 Conclusions

We conducted an in-depth analysis of Wikipedia to shed some light on how real-world events such as political conflicts, natural catastrophes, and new scientific findings are mirrored by article updates in Wikipedia. To this end, we gathered and annotated random samples from Wikipedia updates as well as samples obtained using various filters, in order to investigate different characteristics of the Wikipedia Edit History. We found that events are correlated with bursts of edits, identified connections between events and language as well as meta annotations of updates, and showed that temporal information in edit content and from timestamps can provide clues on the event-relatedness of updates. The results of our experiments on automatic extraction and summarization of events from Wikipedia updates are promising, with possible applications including the construction of entity-specific, annotated timelines and news tickers.

We presented Wikipedia Edit Reporter, a Web-based system for generating temporal summarization of real-world events such as political conflicts, natural catastrophes, and new scientific findings that are mirrored by article updates in Wikipedia. Our system helps users explore the temporal development of events for entities of interest, by presenting an annotated timeline and a concise summarization. Moreover, it is able to find historical information about events that are no longer available in the current version of Wikipedia, giving the user a comprehensive view of the event, and not only the socially accepted final interpretation of the event and its implications. We presented demonstrated that our system is capable of automatic extracting and generating temporal summarization of events from Wikipedia updates enhancing real-world applications, such as, entity-specific, annotated timelines and news tickers.

Regarding future work, we plan to study opinions and controversies that occur in the context of event-related updates in Wikipedia. We think that this can be useful for providing users with more comprehensive overviews covering different schools of thought and points of view. More advanced linguistic and stylistic features of updates might be leveraged to improve classification and clustering. Finally, updates and discussions can lead to further insights on social relationships between users (friendship, rivalry, etc.), and provide clues about the provenance of event-related information contributed by different users.

4 Extraction of Dynamic Event Structures from Wikipedia

Detecting dynamic relationships and associated events poses multiple interesting technical challenges. First, these relationships do not conform to any pre-existing schema and therefore can not be discovered by leveraging language patterns as in previous works on static relationship extraction. Second, the underlying events often have a flexible timeline that is hard to know a priori, e.g., one event can last for a short time over a week, while another could last over several weeks. Third, the entities display a great deal of flexibility in their participation in the underlying events, mainly reflected in the number of participants (some events can involve two entities while others are among several entities [DSJY11]). Fourth, as a real-life event happens, the Web community mobilizes itself to report that. Some information generated in a particular time period will no longer be available in a future version of the articles of the entities involved in the event. Thus, it is important to provide users the possibility to access historical information, giving a comprehensive evolution-aware entity-based view.

A dynamic and complex event structure extracted from Wikipedia can be valuable in many applications. For example, it can be used in recommendation scenarios, where users interested in specific entities get suggested with related entities involved in common events. It can also be used to facilitate knowledge management research (for instance, ontology construction and alignment) by introducing the dynamic structure of entities in temporal dimension. Understanding the evolution of entity involvement during the course of an event can also help to gain insights into collective attention to the entities, which benefits several analytic applications such as advertisements.

In this chapter we introduce a general model which is agnostic to linguistic constraints. Furthermore, we establish a new methodology for detecting events based on explicit relationships identification. To this end we adapt, formalize, and improve the dynamic relationship and event mining problem to the Wikipedia domain. We introduce the temporal aspect as a fundamental dimension to enrich content with semantic information via historical user edits.

We conduct comprehensive experiments on a dataset of 1.8 *million* Wikipedia articles to show the effectiveness of our proposed solution. Our results demonstrate that we are able to achieve a precision of 70% when evaluated using manually annotated data. Moreover, we compare with the well established Wikipedia's Current Event Portal and find that our proposed methods are complementary.

4.1 Approach

Keeping in mind the goal of detecting events from Wikipedia users' Edit History, we model an event through its participating entities: one event consists of a set of entities that are connected at a given time period. For example, the event "83rd Academy Best Actor Awards" on Jan 25, 2011, can be described by its nominees and winners "Colin Firth", "Jeff Bridges", "James Franco", etc. This way of representing events has the benefit of being agnostic to linguistic constraints of a certain language. Following previous work, we represent an entity by a Wikipedia article using its unique identifier.

4.1.1 Problem Formalization

In this section, we present a formal model which constitutes the basis of our work.

Data Model. In our model, time is represented as an infinite series of time points $\boldsymbol{\tau} := \{\tau_i : \tau_i = nP, n \in \mathbb{N}^+\}$, where P is a time interval unit (e.g., second, day, week). For brevity, we just use the index i to refer to a given time point τ_i. We also consider an *entity collection* E derived from Wikipedia, where each entity is associated to a Wikipedia article. Entity $e \in E$ at a given time point i is represented as a textual document $d_e^{(i)}$, which is the revision of the Wikipedia article at a latest timestamp t before i. Given such assumptions, we further define the *edit* of e at the time point i as $m_e^{(i)} := d_e^{(i)} - d_e^{(i-1)}$, the differences betweeon the document at the current time point ant the previous one, and the *edit volume* as the number of revisions between two time points, $v_e^{(i)} := |\{t : \tau_{i-1} \leq t < \tau_i\}|$.

Dynamic Relationships. A dynamic relationship is a tuple $r := (e_1, e_2, i)$, where $e_1, e_2 \in E$ are the entities for which r holds, and $i \in \boldsymbol{\tau}$ is the time when r is valid. Dynamic relationships can be of two types: *explicit* and *implicit*. They are can be identified according to different strategies.

Events. We define an *event* v as a tuple $v := (E_v, \boldsymbol{\tau}_v)$, where $E_v \subset E$ is the *representative entity set*, i.e. those entities that participated to v and $\boldsymbol{\tau}_v := \{i : i \in \boldsymbol{\tau}, i_{start} \leq i \leq i_{end}\}$ is the *time period* when v occurred.

Problem statement. Given an entity set E, a time window $\boldsymbol{W} \subseteq \boldsymbol{\tau}$, detect any event $v = (E_v, \boldsymbol{\tau}_v)$ such that $E_v \subset E$ and $\boldsymbol{\tau}_v \subset \boldsymbol{\tau}$.

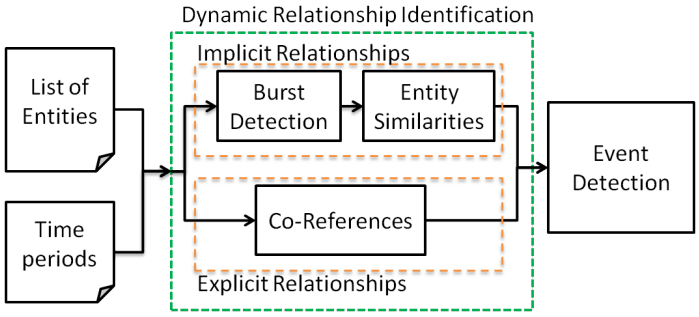


Figure 4.1 WikipEvent architecture for identifying events and relationships between involved entities.

4.1.2 Workflow

The detailed workflow of our system is described in Figure 4.1. In short, our system consists of two steps: *Dynamic Relationship Identification* and *Event Detection*. Given a set of entities and a time period as input, our system provides as output a set of events in which the entities were involved, together with the relationships between them. Such relationships, together with the specified time period, will enable users to fully interpret the detected events (e.g., causes and effects).

The first phase computes the dynamic relationships between entities, using one of the two strategies: *Explicit Relationships Identification* (Section 4.2.1) or *Implicit Relationships Identification*(Section 4.2.2). The former strategy uses explicit links in the articles to establish the relationship between two entities. In a revision, each new or updated link referring to another article indicates explicitly a bind between the source and destination articles, and their corresponding entities. The later strategy is based on two steps. First, *Burst Detection* detects salient bursts of activity in the Edit History, producing a set of pairs of entities that have bursts in the same time. Then *Entity Similarity* step employs a variety of methods to measure similarity of their edits, that are then used to aggregate the pairs of entities to build the co-burst graph for each individual time point. The second phase, generates events described by representative entities and time intervals of involvement. It first builds a sequence of graphs, each one capturing the entity relationships at an individual time point, and then it incrementally builds the connected components grouping entities that are highly related in consecutive time points.

4.2 Relationship Identification

In this section, we present two strategies to create dynamic relationships, corresponding to the first step in our proposed workflow. As we aim to identify events consisting

of a set of entities that are connected at a given time period, it is crucial to define the notion of entity relationships based on an event. Furthermore, such relationships must capture well the temporal dynamics of entities in Wikipedia, where information are constantly added or updated over time. We adopt two strategies to identify entity relationships: (i) we establish a strategy based on *Explicit Relationships Identification*; (ii) we adjust to our domain the *Implicit Relationship Identification* strategy described in [DSJY11], where we adapt the proposed point-wise mutual information (PMI), as well as propose several classes of similarity measures to estimate the confidence of each implicit relationship.

4.2.1 Explicit Relationships Identification

This strategy uses links between Wikipedia articles to establish the relationship between their corresponding entities. The intuition behind is that each link newly added or updated in each article revision indicates explicitly a tie between the source and destination entities. For example, during the Egypt Revolution 2011, the Wikipedia article "Hosni_Mubarak" admits many revisions published. In many of them, the link to the article "Tahrir_Square" is added or refined several times. This reveals a strong relationship between the two corresponding entities with respect to the revolution.

We will further refer to our established strategy as *Co-References*. The dynamic relationships in this case are defined as follows. For each entity e and an edit $m_e^{(i)}$ at time i, if $e_2 \in m_e^{(i)}$ then this implies that there exists a link to the Wikipedia article of the entity e_2 in the content of $m_e^{(i)}$. A relationship between e_1 and e_2 is established if we have links in both directions (from e_1's edit to e_2 and vice versa).

Definition 1. *Given two entities e_1, e_2 and a time interval $I = [i, i+\delta]$, an explicit dynamic relationship between e_1 and e_2 at time point i is a tuple $r_{exp} \in E \times E \times \tau$ such that $r_{exp}(e_1, e_2, i)$ iff $\exists j, k \in [i, i+\delta], e_1 \in m_{e_2}^{(j)}$ and $e_2 \in m_{e_1}^{(k)}$.*

Intuitively, an explicit dynamic relationship captures the mutual references between two entity edits that are made at close time points. The parameter δ accounts for the possible time delay when adding links between two entities. As an example, while the entities "Cairo" and "Hosni Mubarak" are explicitly related during the Egypt revolution in 2011, the two mutual references can be added at different (close) time points; for instance the link from the article of "Hosni Mubarak" to "Cairo" can be added first, and the inverse link can be added one day after.

4.2.2 Implicit Relationships Identification

This strategy infers the relationship between entities through indirect signals, namely burst patterns. Existing work suggests that Wikipedia article view or article edit statistics follow bursty patterns, with spikes driven by real-world events of the entities [CN10, GKK+13]. The intuition behind is that two entities are highly correlated

with respect to an event if their social activities (view and edit) burst around the same time period (co-burst). To avoid the coincidence of two independent entities which burst around the same time period by chance, we further impose that the entities must share sufficient textual or structure similarities during time period of study.

We adapted to the Wikipedia domain the strategy proposed in [DSJY11]; in this adapted approach, we identify a relationship between two entities when their edit histories exhibit bursts in the same or overlapped time intervals (co-burst). The bursts are detected through the *Burst Detection* component. A burst of an entity is the time interval in which the edit volume of the entity is significantly higher than the preceding and following volumes.

4.2.3 Burst Detection

We define a burst of an entity as follows: given an entity e and a time window $\boldsymbol{W}$, we construct a time series $\boldsymbol{v}_e := [v_e^{(i_0)}, \ldots, v_e^{(i_f)}]$ containing the edit volume of entity e at every time point $i \in \boldsymbol{\tau}$. A burst $\boldsymbol{b}_e$ is a sequence of time points $\boldsymbol{b}_e := [i, i+1, \ldots, i+k]$ for which the edit volumes of e are *significantly* higher than the edit volumes observed in neighbouring time points: $v_e^{(i-1)} \ll v_e^{(j)}$ and $v_e^{(j)} \gg v_e^{(i+k+1)} \forall j = i, i+1, \ldots, i+k$. We say two entities e_1 and e_2 *co-burst* at time i iff $i \in \boldsymbol{b}_{e_1}$ and $i \in \boldsymbol{b}_{e_2}$.

For the burst detection, we employ the Kleinberg model [Kle02], a generative method to detect bursts based on finite automata simulations where each automaton corresponds to a state of emitting information at a certain speed. The model identifies bursts as the transitions from one slow speed state to one high speed state.

4.2.4 Entity Similarity

A *co-burst* merely identifies entities that admit high volume of edits at the same time. In practice, two entities can have bursts in the same time interval, even if they are of little relevance. To remedy this, we further assume that the edits of two entities must share sufficient resemblance, for their relationship to be confirmed. As already mentioned, we adapted the proposed entity semantic similarity of [DSJY11], i.e. PMI, and defined three further metrics. Let us denote the similarity of two entities e_1 and e_2 at time i as $S_{method}(e_1, e_2, i)$, then we have the following similarities:

1. *Textual*: This measures how close two entities are in a given time by comparing the plain text content of their corresponding edits. We construct the bags of words $\boldsymbol{bw}_{e_1}^{(i)}$ and $\boldsymbol{bw}_{e_2}^{(i)}$, and use Jaccard index to measure the similarity: $S_T(e_1, e_2, i) := J(\mathbf{bw}_{e_1}^{(i)}, \mathbf{bw}_{e_2}^{(i)})$.

2. *Entity*: This is similar to textual similarity, but with the *bag of entities* $\boldsymbol{be}_e^{(i)}$ (entities that are linked from the edit): $S_E(e_1, e_2, i) := J(\boldsymbol{be}_{e_1}^{(i)}, \boldsymbol{be}_{e_2}^{(i)})$.

3. *Ancestor*: This measures how close two entities are in terms of their semantic types. For each entity, we use an ontological knowledge base where the entity is registered, and extract all ancestors of the entity (entities that are connected to through a *subsumption* relation). Given $\boldsymbol{be}_e^{(i)}$, a *bag of ancestors* $\boldsymbol{ba}_e^{(i)}$ is filled with the ancestors of every entity in $\boldsymbol{be}_e^{(i)}$. We then measure the similarity $S_A(e_1, e_2, i)$ by Jaccard index accordingly.

4. *PMI*: This measures how likely two entities co-occur in the edits in all other entities. Given $i \in \boldsymbol{\tau}$ and $e_1, e_2 \in E$, we construct the graph involving all entities linking to e_1 and e_2 from all edits at i. Let $IN(e)^{(i)}$ denote the number of incoming links for e in this graph, we estimate the probability of generating e by $p(e) = \frac{IN(e)^{(i)}}{N^{(i)}}$, with $N^{(i)}$ being the total number of incoming links in the graph at time i. We then computed the link similarity as $S_{PMI}(e_1, e_2, i) := \log \frac{p(e_1,e_2)}{p(e_1)p(e_2)}$.

4.3 Event Detection

Having defined entity relationships, we detect events by building groups of highly related entities, each one representing a unique event. We cast this problem as the subgraph constructing problem, where the graph is built from Wikipedia entities with edges corresponding to entity relationships. It is not trivial to find such connected components, since the graph is highly dynamic, i.e. the edges change as entity relationships evolve over time, new relationships can be established in a given time and dissipate later, when the tie of the two entities get weaker within its respective revisions. For instance, two entities "Barack Obama" and "Mitt Romney" are highly related during the US presidential election 2012, but they rarely correlate long before and after the event. This demands for an algorithm that can handle the temporal dimension in finding graph connected components.

We detect events by identifying their representative entity sets and the corresponding time period, in an incremental approach. For each individual time point $i \in \tau$, we first build a graph that reflects the entity dynamic relationships at i, which we call *temporal graph*. Then, we cluster entities from individual temporal graphs depending on the type of dynamic relationships (explicit or implicit), as discussed in Section 4.3.4. To find the event time period as well as to identify the event's representative entity set, we compare entity clusters of two adjacent temporal graphs, and incrementally merge two clusters if a certain criterion is met.

4.3.1 Temporal Graph and Entity Clustering

Definition 2. *A* temporal graph $G(i)$ *at time* $i \in \tau$ *is an undirected graph* (E, P)*, where* E *is an entity set and* $P = \{(e_1, e_2)|r(e_1, e_2, j)\}$ *is the set of edges defined by dynamic relationships at a time point* $j \in \boldsymbol{I} = [i, i + \delta]$.

In the above definition, the value δ reflects the lag of edit activities between different Wikipedia articles in response to one real-world event. Note that depending on the type of the dynamic relationships, we have two different types of *explicit* and *implicit* temporal graph respectively.

4.3.2 Explicit Temporal Graph Clustering

In an explicit temporal graph, an edge is defined by the relationship $r_{exp}(e_1, e_2, j)$ and it reflects the mutual linking structure of two Wikipedia entities within interval I. From the temporal graph, we identify the set of maximum cliques C to form clusters of entities that are mutually comentioned from i to $i + \delta$. Each maximum clique $c \in C$ represents an event that occurs at i. The choice of cliques in favor of connected components in this case ensures the high coherence of the underlying events encoded in the group of entities. For example, considering three entities "Anne Hathaway", "James Franco" and and "Minute To Win It" during January 2011. The first two entities are connected by the fact that the two actors co-hosted the ceremony of the 83rd Academy Awards, while the second and third entities are connected because James Franco was at that time a co-performer in the show. This forms a connected component, but putting the three entities together reveals no obvious event.

4.3.3 Implicit Temporal Graph Clustering

In an implicit temporal graph, a candidate edge will be established from two entities which co-burst at a time point $j \in \boldsymbol{I} = [i, i+\delta]$. To mitigate the "co-burst by chance" (Section 4.2.4), we define a *maximum* similarity function:

$$S_{max}(e_1, e_2, \boldsymbol{I}) = \max_{j \in [i, i+\delta]} \{S(e_1, e_2, j)\}$$

and create an edge (e_1, e_2) iff $S_{max}(e_1, e_2, \boldsymbol{I}) \geq \theta$. Intuitively θ is the threshold value used to perform a selective pruning preserving only entity pairs with maximum similarities exceeding it. Unlike in an explicit temporal graph, here we relax the entity clustering requirements by representing the events occuring at i as the connected components.This is due to the nature of the implicit dynamic relationships, where two entities e_1 and e_2 that are not directly connected can still co-burst, through an intermediate entity e' during the interval $\boldsymbol{I}$, by following one path in the graph.

4.3.4 Event Identification

To identify an event, we need to form a representative entity set from a number of temporal graphs, as well as to specify the time interval in which the entity set lies in. This entails aggregating entity clusters of temporal graphs at consecutive

time points. For this task we employ a modified version of the algorithm named *Local Temporal Constraint* (*LTC*), proposed in [DSJY11], that detects events in the dynamic programming fashion.

4.4 Experiments and Evaluation

In order to analyze the performances of the proposed methods, we ran experiments on the quantitative (Section 4.4.3) and qualitative (Section 4.4.4) characteristics of our extracted events. For the specific task of detecting event structures in Wikipedia, to the best of our knowledge, there are no existing resources for a comprehensive list of real-world events. So far several and promising attempts for building collections of real-world events have been conducted: one example is Wikipedia Current Event Portal (called WikiPortal hereafter) which tries to collect human generated event descriptions; another example is the YAGO2 Knowledge Base [HSBW12], where every event consists in a relationship over an entity pair. However, these approaches are limited in term of number of events, complexity, and granularity. Moreover, since a comprehensive event repository does not exist, fairly computing recall for event detection methods is infeasible. Thus, we performed a manual evaluation. Detected events were manually assessed by five evaluators who had to decide if they were corresponding to real events. In detail, for each detected event, the annotators were requested to look at all the involved entities and identify a real-world event by examining various web-based sources (e.g., Wikipedia, official home pages, news search, and web search), that best explained the co-occurrence of these entities in the event during the specified time period. For each set of entities, a label was assigned in order to represent a *true* or *false* event. These assessments contributed to measure the accuracy of our methods.

Finally, we conducted an extensive comparative analysis of our work with the well established WikiPortal by analyzing the events described manually by Wikipedia users versus the events detected by our best performing method (Section 4.5).

4.4.1 Dataset

To build our dataset, we chose to use the English Wikipedia. Since Wikipedia also contains articles that do not describe entities (e.g., "List of mathematicians" maintains a list of references to other article, and is not considered as an entity article), we selected Wikipedia articles corresponding to entities registered in YAGO2 and belonging to one of the following classes: person, location, artifact, or group. In total we used for our study $1,843,665$ articles, each corresponding to one entity. Furthermore, for our study we chose a time period ranging from the 18^{th} January 2011 until the 9^{th} February 2011. This period covers important real-world events such as the Egypt Revolution, the Academy Awards, the Australian Open, etc. The choice of a rela-

tively short time span simplifies the manual evaluation of the detected events. Since using days as time units has been shown to effectively capture the news-related events in both social media and newswire platforms [BAH12], we used the day granularity when sampling the time. We name the whole dataset, containing all the articles, as *Dataset A*. Furthermore, we created a sample set, called *Dataset B*, by selecting entities that were actively edited (more than 50 times) in our time period. The intuition behind this selection is that a large number of edits is more likely to be caused by an event. Consequently, this sample contains just 3,837 Wikipedia articles.

4.4.2 Implementation Details

To store the whole Wikipedia Edit History dump and to identify the edits, we made use of the JWPL Wikipedia Revision Toolkit [FZG11b]. JWPL solves the problem of storing the entire Edit History of Wikipedia by computing and storing differences between two revisions.

To resolve the ancestors of a given entity, we employed the YAGO2 knowledge base [HSBW12], an ontology that was built from Wikipedia infoboxes and combined with Wordnet and GeoNames to obtain 10 million entities and 120 million facts. We followed facts with *subClassOf* and *typeOf* predicates to extract ancestors of entities. We limited the extraction to three levels, since we observed that going to a higher level included several extremely abstract classes (such as "Living people"). This lowered the discriminating performance of the similarity measurement.

For burst detection we implemented Kleinberg's algorithm using the modified version of CShell toolkit [1]. We set the density scaling to 1.5, the transition cost to 1.0, and the default number of burst states to 3 (for more details, refer to [Kle02]). We observed that changing parameters of the burst detection did not affect the order of performance between different event detection methods. For the dynamic relationships, we set the time lag parameter δ to 7 days and γ to 0.8, as these values yielded the most intuitive results in our experiments.

4.4.3 Quantitative Analysis

The goal of this section is to numerically evaluate our approach under different metrics: (i) total number of detected events, and (ii) the precision, i.e. the percentage of *true* events. For the parameter selection, note that the graph created based on the explicit strategy does not have any weights on its edges. On the contrary, the implicit strategy creates a weighted graph based on the similarities, and the temporal graph clustering depends on the threshold θ to filter out entity pairs of low maximum similarity. We varied the value of θ and noticed that lowering it resulted in a larger number of entity pairs that coalesced into a low number of large events. These events

[1] http://wiki.cns.iu.edu/display/CISHELL/Burst+Detection

Table 4.1 Performance on Dataset A.

Strategy	Method	Events	Precision
Explicit	Co-References	186	70%
Implicit	PMI	124	39%
	Ancestors	33	51%
	Entities	21	62%
	Textual	78	1%

Table 4.2 Performance on Dataset B.

Strategy	Method	Events	Precision
Explicit	Co-References	120	80%
Implicit	PMI	80	69%
	Ancestors	18	50%
	Entities	12	60%
	Textual	15	7%

containing a large number of various entities could not have been identified as real events. Therefore, for the following experiments we used $\theta = 1$.

We evaluate approaches for the implicit relationship identification strategies as defined in Section 4.2.2, referred to as the following *methods*: *Textual*, *Entities*, *Ancestors*, and *PMI*, as well as for the explicit strategy as defined in Section 4.2.1, referred to as the *Co-References* method. The results are presented in Table 4.1 and Table 4.2. The number of events detected for the different similarity setups is presented in the third column of the tables. As expected, we detect more events in *Dataset A* as in *Dataset B*, due to the higher number of entities taken into consideration. The biggest number of detected events is provided by *Co-references* in both datasets. This is attributed to the parameter-free nature of the explicit strategy, while for the implicit strategy, a portion of events are removed by a threshold. Comparing the methods used by the implicit strategy, *PMI* detects more events than any other method. This is caused by the difference in computing the entity similarity $S(e_1, e_2, t)$. *PMI* considers the sets of incoming links, that account for relevant feedback to our e_1 and e_2 from all the other entities in Wikipedia. This results in more entity pairs, and more clearly defined and coherent events, while the other implicit strategy methods tend to conglomerate most of the entities in larger but fewer events. *Textual*, *Entities*, and *Ancestors* compute $S(e_1, e_2, t)$ starting from the edited contents of two entities at a given time. A large amount of content concerning entities that are not explicitly referring to e_1 and e_2 will be taken in consideration as well, making the value of $S(e_1, e_2, t)$ lower. Therefore, using the same value for θ as for the *PMI*, produces a lower number of entity pairs, and consequently of detected events.

The precision of every setup, i.e. the percentage of true detected events, is sum-

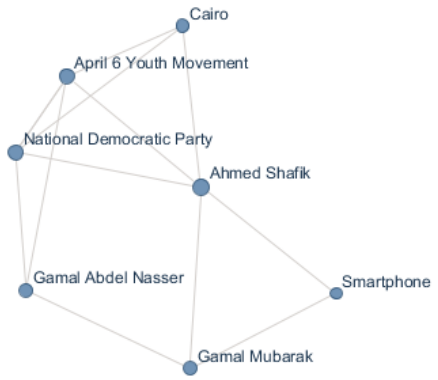


Figure 4.2 Example I: Relationships identified for the event known as "The Friday of Anger" in the context of the 2011 Egyptian Revolution.

marized in the fourth column of Table 4.1 and Table 4.2. Among the implicit strategy methods, we notice clear a benefit of using similarities that take semantics into account (*Entities*, *Ancestors*, and *PMI*) over the string similarity (*Textual*). *Ancestors* performs worse than *Entities* in both datasets, showing that the addition of the ancestor entities introduces more noise instead of clarifying the relationships between the edited entities. *Entities* achieves similar performances on both datasets. *PMI* achieves better performance in *Dataset B* than the other implicit similarities since it is leveraging the structure of incoming-outgoing links between Wikipedia articles. However, *PMI* performs worse on *Dataset A* due to the higher number of inactive entities considered, introducing noisy links.

Finally, *Co-References* outperforms all the implicit strategy methods on both datasets, showing that a clear and direct reciprocal mention is stronger than similarities inferred from the text of the edit. Generally, all methods performed better or comparable on *Dataset B* in comparison to *Dataset A*. This shows that selecting only the entities that are edited more often improves the quality of our methods. Although less events are detected in *Dataset B*, more of them correspond to real life events.

4.4.4 Qualitative Analysis

In this section, we do a qualitative evaluation of the events identified in *Dataset A*. First, we focus on and describe some of the events detected by our best method *Co-References* (highlighted in Section 4.4.3). Second, we analyze some cases where our methods failed, proposing the causes.

In Table 4.4 we present and discuss some events identified by our best method *Co-References* matching real-world events. For each detected event, we report the entities involved, the time when the event occurred, and a human description of the event extracted from web-based sources.

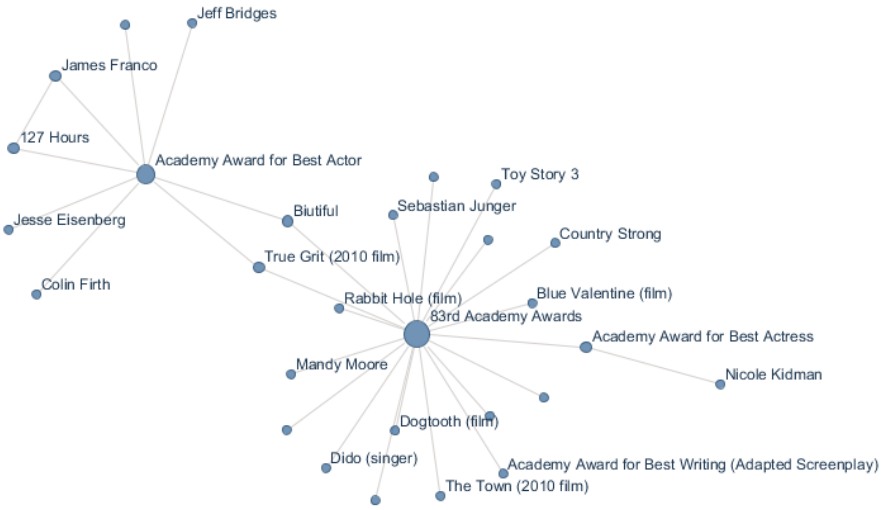


Figure 4.3 Example II: The entities and the relationships identified for the 83rd Academy Awards Nominations.

Moreover, we show the graph structure of two good examples of identified events:

Example I. We depict the relationships associated to the real-world event known as the Friday of Anger, in the context of the Egyptian Revolution in Figure 4.2. On January 28, 2011, tens of thousands filled the streets across Egypt, to protest against the government. One of the major demonstrations took place in *Cairo*. The organization of the protests was done with the help of internet and *smartphones*, and some of the organizers were the *April 6 Youth Movement* and the *National Democratic Party*. Protesters held portraits of former president *Gamal Abdel Nasser*. The aviation minister and former chief of Air Staff *Ahmed Shafiq*, as well as *Gamal Mubarak*, were seen by the government as probable successors of Hosni Mubarak.

Example II. The graph structure of another outstanding detected real world: the announcement of the nominees for the 83rd Academy Awards, on January 25, 2011, is shown in Figure 4.3. We can see as a the biggest central node the *83rd Academy Awards*, and as a secondary node the *Academy Award for Best Actor*, having as connecting nodes *True Grit* and *Biutiful*, which were nominated for more categories.

Finally, in Table 4.3, we report some failures of our methods to identify real-world events, together with the causes that lead to such erroneous output. Depending on the method, we can notice different patterns that cause false positives. The *entity-based similarity* usually fails because of updates containing a large number of common entities that are not involved in any common event. Using the *ancestor-based similarity* can provide false events because some entities that are very similar, and share

a large number of ancestors, have coincidentally concurrent edit peaks in the same period. The *PMI* fails because of similar causes: entities that share a lot of common incoming links to the entities contained in the edits done on the same day. Finally, the *Co-References* method seems to fail when the reciprocal mentions originate in relationships that are independent of any event.

4.5 Comparative Analysis

Finally, we compare events detected by our approach with events present in WikiPortal in the same time period. Users in WikiPortal publish a short description of an event in response to the occurrence of a real-world happening and annotate the entity mentions with the corresponding Wikipedia articles. The event descriptions can also be grouped into bigger "stories" (such as Egypt revolution), or can be organized in different categories such as sports, disasters, etc.

We conducted the comparison by considering the 130 (70% of 186) true events detected with the *Co-References* method on *Dataset A*, since this is the setup that gave the largest set of true events.

Then, we collected 561 events from WikiPortal within the period of interest by considering all event descriptions inside the same story as representing a single event. We further considered only those event descriptions annotated with at least an entity contained within *Dataset A*, getting in total 505 events. In principle, these events can be detected by our method.

In order to assess the overlap between the two event sets, we classify events according to the following categories:

1. *Green:* The event in one set, with all its participating entities, is present in the other event set either as a single event or as multiple events.

2. *Yellow:* The event is partially present in the other event set, i.e. only a subset of its participating entities appears in one or more events in the other set.

3. *Red:* The event is not reported in the other event set.

We provide explanatory examples for each category in Table 4.4, along with explanations of each classification choice. We can observe 2 patterns for the events in our set belonging to the green category: (i) one event in *Co-References* is spread over different events in WikiPortal; for instance, the event regarding the candidacies for the Fianna Fail party is reported in WikiPortal through different events, each one focusing on a single candidacy; (ii) one event in *Co-References* corresponds to one event in WikiPortal. The yellow category generally covers the case where an event represents a non mentioned aspect of an event in WikiPortal. For instance, for the Australian Open tennis tournament only the men's semi-final and final matches are

Table 4.3 False positive events, along with the probable reason why our methods failed.

Entities	Method	Cause
Alexis Korner, Fleetwood Mac, Bob Brunning	Co-References	Bob Brunning was a member of the Fleetwood Mac a British-American rock band , and Alexis Korner wrote a book about them. They were not involved in any common events in our period.
Alexa Nikolas, Ariana Grande	PMI	Two different persons, that just look alike, share some of their own incoming links, but not much more. They were not involved in a common event in our period, but their articles might have experienced unrelated edit peaks simultaneously.
Saudi Arabia national football team, Ghana national football team, Canada men's national soccer team	Ancestors	The entities have a lot of common ancestors, coming from the Sports domain, and all of them had peaks of activity in the same time. However, they were not involved in common events during the studied period.
Tura Satana, Barack Obama	Entities	Tura Satana died, but Barack Obama did not have any connection to her in the period under investigation, although both entities experienced edit peaks and the entities contained in the edits were similar.

reported in WikiPortal, without mentioning the other matches, which are reported in *Co-References*. Similarly, the Friday of Anger (in the context of the Egyptian revolution) and the Academy Awards nominations are present in WikiPortal, but our detected events are endowed with more entities that do not appear in the portal. Finally, the red category collects those events that are not reported in WikiPortal at all, like the Royal Rumble wrestling match.

We noticed that 60% of the events detected by *Co-References* are present fully or partially in the WikiPortal. For the sake of clarity, in Table 4.5 we present some of the events that are present in WikiPortal but our method was unable to detect, along with an explanation. The main patterns are: (i) the events involve just one entity; (ii) the events involve entities that are highly unlikely to reference each other because of their different roles in the common events.

In conclusion, WikiPortal and *Co-References* can be seen as complementary methods for event detection, each with its own advantages and disadvantages. While WikiPortal is user-contributed and requires human effort to curate events, our method is fully unsupervised, and it can detect additional events minimizing the human intervention.

4.6 Conclusions

In this chapter, we introduced the temporal aspect as a fundamental dimension for enriching content with semantic information, and provided a model which can capture dynamic event structures. Focusing on Wikipedia, we are able to find historical information, events.

Because of the specific task we consider, no annotated collections were available,

Table 4.4 Co-References extracted events matching real-life events, along with their date and a human description.

Entities	Date (2011)	Human Description	Category Explanation
Category: Green			
Brian Cowen, Michel Martin, Mary Hanafin, Mary Coughlan(politician), Fianna Fáil	From Jan 18 to Jan 26	The Irish PM Brian Cowen announced his stepping down as leader of the ruling Fianna Fail party, and different candidacies for the leadership follow his decision.	The event is globally reported in WikiPortal through different daily events.
Saad Hariri, Najib Mikati	Jan 25	Supporters of Lebanese caretaker Prime Minister Saad Hariri call for a day of protests following Hezbollah's support for Najib Mikati as Prime Minister	The entities participating to the event are all mentioned in an event within WikiPortal.
Category: Yellow			
Gamal Abdel Nasser, Ahmed Shafik, Smartphone, Cairo, April 6 Youth Movement, Gamal Mubarak, National Democratic Party	Jan 28	In the context of the Egyptian Revolution, the Friday of Anger takes place: tens of thousands filled the streets across Egypt, to protest against the government.	Most of the entities appear in WikiPortal within different events. The entities *Gamal Mubarak* and *Gamal Abdel Nasser* do not appear.
Li Na, Kim Clijsters	Jan 19	Australian Open 2011 women's final: Li Na vs Kim Clijsters.	The Australian Open is mentioned two times, but always focusing on men's matches. The women's final is not reported.
Andy Murray, Illya Marchenko	Jan 20	Andy Murray secured a place in the third round of the Australian Open after a routine victory over Ukrainian Illya Marchenko.	The event is not present in WikiPortal. However, it is reported that Andy Murray was defeated by Novak Djokovic during the final match.
Dallas, Donald Driver, Charles Woodson	Feb 6	Green Bay Packers won the Super Bowl XLV in Dallas, although the players Charles Woodson and Donald Driver suffered injuries.	The Super Bowl XLV is mentioned in WikiPortal, but there are no references to the injuries of the two players.
James Franco, Colin Firth, Academy Award for Best Actor, Biutiful, True Grit, 83rd Academy Awards, ... (Figure 4.3)	Jan 25	Announcement of the nominees for the 83rd Academy Awards	The event is reported in WikiPortal, but few participating entities are mentioned.
Category: Red			
Vickie Guerrero, Hornswoggle, Layla El, Dolph Ziggler, Booker T, Professional wrestling, Kane, Santino Marella	Jan 30	The 2011 Royal Rumble organized by WWE takes place, involving a lot of wrestlers.	The event and no one of its entities are reported in WikiPortal.
Ashley Young, Charlie Adam	Last week of Jan	Liverpool Football team planned improved offers for Ashley Young and Charlie Adam.	The event and no one of its entities are reported in WikiPortal.
Silent Witness, Bruce Forsyth, Loose Women	Jan 26	In the context of the 16th National Television Awards, presented by Bruce Forsyth, Loose Women and Silent witness are nominated.	The event and no one of its entities are reported in WikiPortal.
Catwoman, The Dark Knight, Bane	Jan 19	Warner Bros. Pictures announced that Anne Hathaway has been cast as Catwoman and Tom Hardy as Bane in "The Dark Knight Rises".	The event and no one of its entities are reported in WikiPortal.
Laura Schlessinger, Michele Bachmann	Jan 19	On the Sean Hannity Show, Dr. Laura Schlessinger and Michele Bachmann discuss the left misconstrued and attacked comments of Conservative women.	The event and no one of its entities are reported in WikiPortal.
Sean Parker, Eduardo Saverin	Feb 1	Sean Parker and Eduardo Saverin sell some of their Facebook Stocks after the company's public listing	The event and no one of its entities are reported in WikiPortal.

Table 4.5 Examples of events from WikiPortal that were not detected by our method, Co-References.

Event Description	Date(2011)	Explanation
Apple records record profits of $6 billion as consumers consumed more of its products than was thought (BBC)	Jan 18	The event involves just one entity
Chinese President Hu Jintao begins a four-day state visit to the United States.	Jan 18	It is highly unlikely that the prominent entities have mentioned each other
Exotic birds are found to have been driven into Britain's back gardens by the extreme cold, as more than half a million people participate in the largest wildlife survey in the world	Jan 29	It is highly unlikely that the event attracted the attention of the Wikipedia community
Russia starts a search for a missing military satellite launched into the wrong orbit.	Feb 1	The event involves just one entity
Researchers report that fishing rates in the Arctic are 75 times higher than those reported by the U.N., suggesting future increased exploitation is less possible than previously thought.	Feb 4	It is highly unlikely that the prominent entities have mentioned each other

thus we manually assessed the performance of our methods using a data set of 1.8 million articles. Over an extensive set of experiments we established the effectiveness of out proposed approach and investigated different strategies and methods. We have shown that an explicit relationship identification strategy performs better than an implicit one, achieving a maximum precision of 70%. We observed a further increase to 80% in precision when using only actively edited articles. We demonstrated the effectiveness of our approach in exploiting Wikipedia edits in order to detect relevant events.

We further conducted a comparison between events detected by our methods, using the *Co-References* approach, with events present in WikiPortal in the same time period, highlighting that they can be seen as complementary sources of events. The former is user-contributed and requires human effort to curate events, the latter is fully unsupervised, and it can detect additional events without any human intervention.

The events that encompass the dynamic relationships discovered with our methods are currently summarized only using the time period and the entities involved, a straightforward extension to our work is to provide a textual description to the event, that would offer the user a comprehensive view on the dynamics of the event and the involved entities. Based on the current approach for detecting dynamic relationships, we will develop in the future a probabilistic model for temporal retrieval and ranking, which takes into account the temporal dimension, entity/event mappings, user involvement, as well as the dynamic multi-relational graphs from social networks.

5 Soliciting Crowd Wisdom via Crowdsourcing

5.1 Motivation

Supervised machine learning needs annotated data to learn from. Traditionally, these annotations are done by experts, at a high cost. Lately, a rising trend for obtaining the annotations by means of crowdsourcing can be observed in the literature. Crowdsourced annotations, although having advantages, also pose new problems, such as their questionable reliability and quality. Usually the crowd workers are paid symbolic amounts, and the tasks they solve fail to provide additional motivation such as entertainment. Therefore, usually the crowd workers do not provide high quality labels, because of low motivation or lack of skill. To tackle the annotation quality assurance problem, we propose to employ an EM-based algorithm to simultaneously find the hidden underlying labels and evaluate the workers. An efficient way to gather annotated data is by using active learning, by specifically requesting annotations that would improve the algorithm. Active learning can also benefit from crowdsourcing, nevertheless this poses new challenges compared with traditional machine learning. We identify and tackle these challenges by proposing an integrated framework and try to answer questions such as: what kind of new instances selection strategies could be used, what kind of algorithms are more suitable for actively learning from the crowd, what is the optimal allocation of resources for a round of active learning? Thus, we identify and address some of the challenges raised by employing crowdsourcing in order to gather annotated data to be used in machine learning.

Using crowdsourcing for gathering labels can be beneficial for supervised machine learning, if done in the right way. Crowdsourcing is more cost-effective and faster than employing experts for labeling the items needed as training examples. Unfortunately, the crowd produced labels are not always of a comparable quality. Therefore, different methods could be employed in order to assure label quality. One of them is redundancy, by gathering more than one label per item, from different assessors. In our work we introduce a novel method for aggregating multiple crowdsourced binary

labels, taking into account the worker's history and how well the worker agrees with the aggregated label. According to previously solved tasks, the worker expertise, or the confidence we have in his labels can be assessed. The computation of the aggregated crowd label is mutually reinforced by the assessment of the worker confidence. Besides a method for computing a hard nominal aggregated label, we also propose a soft label as an indicator of how much the labelers agree and how strong their labels are. Furthermore, we investigate whether worker confidence should depend on the provided label, whether discriminating between positive and negative answer quality can be beneficial. We evaluate our method on multiple datasets, covering different domains and label gathering strategies. Moreover, we compare against other state of the art methods, showing the effectiveness of our proposed approach.

In Chapter 6 we introduce our novel method for aggregation of different crowdsourced labels, by taking into account the worker expertise, expressed as the confidence we have in the provided labels. Furthermore, we assess of different ways of computing the worker confidence, as well as various ways of incorporating them in the computation of the final aggregated label. Moreover, we do a comprehensive evaluation of the effectiveness of our method on different datasets and comparison with other state-of-the art methods.

As crowdsourcing has lately become ubiquitous in machine learning as a cost effective method to gather training labels, in the second part of this chapter, we examine the challenges that appear when employing crowdsourcing for active learning, in an integrated environment where an automatic method and human labelers collaborate. Computers are adept at solving complex computational tasks, however there are a wide range of tasks at which humans are better. Tasks requiring subjective, perceptual, emotional or intellectual discretion capacities are some examples. The paradigm of social computation was developed specifically for solving these types of tasks. We tackle the problems that arise in building an integrated system where humans and computers work together towards improving their performance on the task at hand and show that crowdsourcing coupled with machine learning can be a cost effective solution to alleviate specific problems where machines alone would fail, and employing only humans would prove to be too expensive. By using active learning techniques on crowd-labeled data, we optimize the performance of the automatic method, while keeping the costs low by gathering data on demand. In order to verify our proposed methods, we apply them to the task of deduplication of publications in a digital library by examining metadata. We investigate the problems created by noisy labels produced by the crowd and explore methods to aggregate them. We analyze how different automatic methods are affected by the quantity and quality of the allocated resources as well as the instance selection strategies for each active learning round, aiming towards attaining a balance between cost and performance.

In Chapter 7 we propose a novel integrated framework for active learning using crowd assigned labels, and we identify the major challenges that can arise when deploying such a framework. Furthermore, we provide extensive experiments using

various automatic methods that learn to perform a well-defined task by exploiting the the wisdom of the crowds. Finally, we envision further extensions that would make the proposed approach more robust.

5.2 Related Work

Human computation [vA09b, YCKK09] is a computer science technique in which a machine outsources certain steps to humans aiming towards a symbiotic human-computer interaction. One of the most popular means for facilitating human computation is crowdsourcing. Crowdsourcing either motivates participation with rewards or it relies on the altruism of the participants by bringing the larger purpose to light. For example, [BBC12] proposes to use existing social applications instead of using a payed workforce to post microtasks that the friends of the user can solve. Also without a financial motivation, Games with a Purpose [VA06] motivate participation by providing entertainment and hide the larger purpose of the system users.

In our work we focus on paid Crowdsourcing, and we leverage Amazon's Mechanical Turk micro task marketplace(MTurk[1]), one of the most well known and used crowdsourcing platforms, for label gathering of multiple labels to be used for supervised learning. The marketplace is named after an 18th century automatic chess-playing machine, which was handily beating humans in chess games. Of course, the robot was not using any artificial intelligence algorithms back then. The secret of the Mechanical Turk machine was a human operator, hidden inside the machine, who was the real intelligence source. An analysis of the Amazon Mechanical Turk Marketplace is provided in [Ipe10]. Some of the insights gained after crawling and analyzing all the available hits for a period of time include: most of the tasks have tiny rewards, the rate of task completion is slightly higher than the rate of arrival, the completion time follows a power law, and the market is heavy-tailed market, in terms of both requester and worker activity, following a log-normal distribution.

The tutorial [IP11] attempts to provide a holistic view of the area of crowdsourced human computation, the main methods and key issues of employing crowdsourcing are outlined. Covering the literature in computer science that focuses on the topic of crowdsourcing, and also areas of research in statistics, control theory, economics, psychology, and epidemiology the tutorial addressed topics such as such as managing quality, task design and workflow design, incentives, game design, and behavioral issues, market design issues, constraints on general purpose infrastructures for human computation, and social and economic impact. Quality control when dealing with noisy crowds is identified one of the main issues of crowdsourcing.

An increasing interest in crowdsourcing can be noticed in a variety of research areas such as in the database community [FKK+11, FFK+11], in the IR community [ABYBY11, KKK+11], or in developing general crowdsourcing algorithms [VGMH+12,

[1] http://www.mturk.com

MWK+11a]. Crowdsourced marketplaces have made it easy to recruit a crowd of people for performing tasks that are difficult for computers(such as performing entity resolution [WKFF12, BIPR12]) that can be thought of as database problems, where each item is a row in a table with some missing attributes (labels) that need to be supplied by crowd workers, giving rise to a new generation of database systems, known as crowd-sourced databases [FKK+11, FFK+11, MWK+11a, KSKK11, PGMP+12]. Research in this area either focuses on integrating crowdsourcing functionality efficiently into database management systems, or on algorithms suitable for this kind of environment. Systems such as CrowdDB [FKK+11], Qurk [MWK+11b], and Deco [PGMP+12] connect traditional data storage systems with crowdsourcing platforms such as Amazon Mechanical Turk to gain additional information on queries. To provide the same functionality as traditional relational database management systems, research has further focused on optimizing crowd accesses, focusing either on reducing the budget while assuming that the crowd answers perfectly [WKFF12], observing the quality of the answers by the crowd workers as an orthogonal problem, or on addressing fault-tolerance by handling noisy answers from the crowd [GPGM12, MWK+11a, MKM+12]. For example, using two sorting algorithms, QuickSort and BubbleSort as examples, [GK13] illustrate how algorithms handle noise, which measures can be taken to make them more robust, and how these changes to the algorithms modify the budget and quality estimates of the respective algorithm.

By drawing on theory from organizational behavior and distributed computing, [KNB+13] foresees a framework for a complex, collaborative and sustainable future crowd workplace. Research challenges for such an edeavour are outlined in areas such as: workflow, task assignment, hierarchy, real-time response, synchronous collaboration, quality control, crowds guiding AIs, AIs guiding crowds, platforms, job design, reputation and motivation.

We are leveraging the wisdom of the crowd by employing human computation in it's crowdsourcing form in order to gather labels for items that will be used for machine learning.

Crowdsourcing and Quality of Work

Crowdsourcing taps into the wisdom of crowds. It involves posing a hard question to a set of workers and aggregating their individual responses in order to deduce the answer to the question. Rather than aggregating the answers for each question in isolation, state-of-the-art methods investigate the global matrix of user provided answers to all the questions in order to simultaneously elicit both the user reliabilities and the true answers.

Crowdsourcing can be used for machine learning, therefore enabling automatic methods to learn directly form crowds. Unfortunately, distributing labeling work to crowdsourcing platforms, such as Amazon Mechanical Turk, exposes the requester to

quality risks. Verifying the quality each label is an expensive operation that discards many of the advantages of crowdsourcing. A common solution to this challenge is to rely on redundancy and repeated labeling: the same task is completed by multiple workers. Then an aggregation method is used to infer the labels that will be used for learning. Due to the high variance in annotation accuracy exhibited by individual crowd workers, often multiple crowd workers are asked to label each example, in order to infer a single consensus label. While simple majority vote computes consensus by equally weighting each worker's vote, weighted voting assigns greater weight to more accurate workers, where accuracy is estimated by inter-annotator agreement or agreement with known expert labels.

Most of the challenges of controlling quality when employing crowdsourcing for machine learning are outlined in [Lea11]. The challenges stem from issues such as the human factors, automation of the quality control process, annotation process and guidelines, the worker and task organization, or the minority voice giving rare insights but hidden in spurious noise. The combination of machine learning and crowdsourcing brings advantages such as: more labeled data is available, more hybrid systems appear, the gathered data that is more uncertain, diverse, specific, ongoing and more rapid, and on-demand evaluations are easier to do; one disadvantage might be that the re-use of data is limited.

It has been shown that acquiring multiple, albeit noisy labels can significantly improve the data quality, and that getting more noisy labels per item and then aggregating them is more accurate than getting more expensive, and hence assumably more accurate, labels [SPI08]. Therefore, selective repetition of some micro tasks can improve data quality. [SPI08] uses only majority voting to aggregate labels from multiple users, and is primarily concerned with identifying the items that will benefit from more labels. However they made a strong assumption that all of the workers were of the same ability, and the proposed strategy requests a relatively large number of repeated labels for each sample. Moreover, not all samples need redundant overlapping labels; it is more effective to use overlapping labels for samples whose overlapping labels show low agreement, and for samples whose overlapping labels bring high uncertainty to a learned model. Their method requires repeatedly labeling of each sample to determine whether to use those overlapping labels for a sample in the training process. Despite their insightful analysis, the practical value of repeated labeling varies greatly with different cost models and with different labeler accuracies.

Nevertheless, unknown differing qualities require more sophisticated strategies to deal with noisy labelers in general. Expanding the investigation of [SPI08] of how labeling effort can be best used to maximize learner accuracy, [KL11] integrates knowledge of annotator accuracy obtained using methods from [SOJN08]. By using simulated experiments they show that labels can be thus aggregated more effectively and thereby the learning rate of a supervised model can be improved. Strategies to collect high quality labels at a low cost for training retrieval models have also been proposed in [YMSM10]. Based on the observation that *urls* are more often judged

as not relevant, and rarely as perfectly relevant to a query, the authors employ the following heuristic if a labeler thinks a *url* is relevant to a given query, it is worthwhile to verify others' opinions, but if a labeler thinks a *url* is bad, his opinion should be trusted. [BGC10] discusses the trade-off between determining consensus annotations and maximizing coverage on the training data, providing two main insights. Firstly, the amount of annotations per example should depend on the level of agreement between annotators. Secondly, although annotator disagreement and classifier uncertainty may be easily confused, classifier uncertainty can be useful to guide annotation, while annotator disagreement can serve as an indicator of poor training data. Because crowdsourced data usually does not fit well with the model used in [SPI08] as the noise is not constant across examples, the trade-off between consensus and coverage should depend on the level of agreement between annotators.

The aggregation of crowd opinons, using majority voting, as well as a machine learning algorithm, was studied using samples of individual responses to IQ tests in [BGMG12]. The authors notice that the aggregated crowd IQ grows quickly with its size but then saturates, indicating diminishing returns from each additional member; the decisions based on the aggregated opinions of homogeneous crowds are better than the decisions based on the crowds' best performing members, whereas the best approach for a heterogeneous population is to identify the best performing individual and base the decision on her opinions. Moreover, an individual contribution to the Crowd IQ is not solely related to the participant's IQ but also depends on the uniqueness of her contribution in the context of a given crowd. Using a similar methodology to examine the crowd on Mechanical Turk, [KBK+12] show that crowds composed of workers of high reputation achieve higher performance than low reputation crowds, the effect of the amount of payment is non-monotone (both paying too much and too little affects performance), and when when the task is designed such that incorrect responses can decrease workers' reputation scores, higher performance is achieved.

If we define the problem of learning from crowds as: given a set of user ratings, collectively determine the reliability of each user and the true quality of each item, we can divide the approaches into two categories: machine-learning based and linear-algebraic based. The machine-learning approaches are based on variants of EM, but provide no guarantees as to how well they perform. Algebraic approaches, on the other hand, can provide theoretical guarantees on the error in estimating item qualities, but so far have been limited to either complete assignment graphs (when each user rates all items) or to random graphs (when the assignment of users to items is random). We mention some of the proposed algebraic approaches, such as [GKM11] relying on a spectral algorithm that provably learns the true item qualities, with bounded error, [KOS11] using belief propagation to derive both a set of user reliabilities and an estimate for item qualities for a sparse random graph, and [DDKR13] proposing an eigenvector-based technique to estimate both the user reliabilities and the item qualities.

Learning from crowds can be further categorized based on the aspect of the infer-

ence target into two groups: one aiming to infer the true labels and the other aiming to infer mainly predictive models. Most of the existing methods are categorized into the former group, while the methods such as [RYZ+10], [YFRD11] and [KTK12] are categorized into the latter group. While [RYZ+10] and [YFRD11] model a classifier as a parameter and the unobserved true labels as latent variables and infer them using the EM algorithm, [KTK12] infers only the models without estimating the true labels. Instead of introducing latent variables to estimate the true labels, in [KTK12] a personal classifier for each of the workers is employed, and a base classifier is estimated by relating it to the personal models. This model takes account into the ability of each worker and the instance difficulty for each worker, and this idea leads to a convex optimization problem.

One well established method to tackle the quality of items subjected to annotations by labelers of various expertise is to employ an EM algorithm to estimate error-rate of labelers as well as the hidden labels can be evaluated [DS79b]. Dawid and Skene (DS) [DS79b] propose an EM approach to estimating the error rates of patients with respect to yes-no classification of medical symptoms. For a given a list of symptoms, the patients (who are known to have a certain disease) identify and mark the symptoms they have. Based on the true symptoms of the disease, the EM algorithm can estimate error rates of the patients. Adapting the original work to the crowdsourcing domain, their method assumes that each worker is associated with an unknown confusion matrix. Each off-diagonal element represents misclassification rate from one class to the other, while the diagonal elements represent the accuracy in each class. According to the observed labels by the workers, the maximum likelihood principle is applied to jointly estimate unobserved true labels and worker confusion matrices. The likelihood function is non-convex, but a local optimum can be obtained by using an Expectation Maximization (EM) algorithm that can be naturally initialized by using majority voting. The algorithm iterates until convergence, following two steps: (1) estimates the correct answer for each task, using labels assigned by multiple workers, accounting for the quality of each worker; and (2) estimates the quality of the workers by comparing the submitted answers to the inferred correct answers. The final output of the DS algorithm is the set of (estimated) correct answers for each task and the confusion matrix for each worker, listing the error probabilities for each worker. From the confusion matrix we can directly measure the overall error rate for each worker as the sum of the non-diagonal elements of the confusion matrix (properly weighted by the priors): this results in a single, scalar value as the quality score for each worker. [LYZ13] provides finite-sample exponential bounds on the error rate (in probability and in expectation) of hyperplane binary labeling rules for the DS. [HCMF+12], applies the DS method for crowdsourcing relevance judgements for IR, in the context of the INEX 2010 Book Search track. In the presence of systematic bias, the measurement of error rate results as in [DS79b] underestimates of the true quality of the worker and in potential incorrect rejections and blocks of legitimate workers. To address this issue, in [IPW10] introduces an algorithm that separates the

unrecoverable error rate from bias. Using the confusion matrix of each worker, every assigned hard label from the worker is transformed into a soft label, which reflects the error rate of the worker. Thus, the uncertainty and cost associated with each soft label can be evaluated, enabling the computation of a quality score for each worker, potentially adjusted for the different costs of the misclassification errors, separating the intrinsic error rate from the bias of the worker, allowing for more reliable quality estimation.

Also by employing an EM algorithm, Raykar et. al [RYZ+10] estimate the error-rates and the underlying hidden labels in the absence of a golden standard, by employing a Bayesian approach and worker priors for each class. Two key assumptions are made: the performance of each annotator does not depend on the feature vector for a given instance, and conditional on the truth the experts are independent. Raykar [RYZ+10] employs Expectation Maximization in an unsupervised algorithm that iteratively establishes a particular gold standard, measures the performance of the annotators given that gold standard, and then refines the gold standard based on the performance measures. The performance of each annotator is measured in terms of the sensitivity (bias toward the positive class) and specificity (bias toward the negative class) with respect to the unknown gold standard. The algorithm automatically discovers the best experts and assigns a higher weight to them. In order to incorporate prior knowledge about each annotator, a beta prior on the sensitivity and specificity is used to derive the maximum-a-posteriori estimate.

ZenCrowd [DDCM12] is a hybrid platform that combines algorithmic matching techniques and human intelligence to link entities using a probabilistic framework for the decision process and quality control. ZenCrowd attempts to improve the automatic results of algorithmic matching techniques by involving crowd workers. The system employs a probabilistic reasoning framework to dynamically assess crowd workers, and to combine their outputs taking into account the results of the algorithmic matching, uniqueness constraints, and identity links from the linked open data cloud. Instead of using heuristics or arbitrary rules, ZenCrowd systematizes the use of probabilistic networks to make sensible decisions about the potential instance matches and entity links. All evidences gathered from both the algorithmic methods and the crowd are fed into a scalable probabilistic store and used to process all entities accordingly. The probabilistic model assumes workers acting independently of the difficulty of the labeling task, and of each other. In [DDCM13] the ZenCrowd system is adapted for large scale linked data integration by leveraging a three-stage blocking technique for obtaining the best possible instance matches while minimizing both computational complexity and latency, in order to identifies entities from natural language text using state-of-the-art techniques to automatically connect them to the linked open data cloud.

GLAD [WWB+09] (Generative model of Labels, Abilities, and Difficulties) also formulates a probabilistic model of the crowd labeling process. It leverages inference methods to simultaneously infer the expertise of each labeler, the difficulty of each

item, and the most probable label for each item. It models worker expertise, as a function of the difficulty of labeling the items. While both annotator competence and example difficulty are modelled, annotator bias is not considered.

To evaluate crowd annotations for natural language tasks, such as textual entailment and word sense disambiguation, [SOJN08] acts in a similar way to [DS79b] and estimates worker confusion matrices by implementing a fully-supervised Naïve Bayes estimation with Laplacian smoothing, to construct a weighted ensemble for consensus labeling in which labels are weighted proportionally to the accuracy of the annotator they come from. The authors compare inter-agreement between annotators for crowds and experts, and propose an approach that uses a small amount of expert-labeled training data in order to correct the individual biases of different non-expert annotators. They recalibrate the worker responses to more closely match expert behavior, and then weigh each workers vote by their log likelihood ratio for their given response. However, full supervision can be costly in expert annotation, and defeat the purpose of crowdsourcing.

Most state-of-the-art models propose a probablistic model and use an unsupervised EM algorithm to jointly estimate worker accuracies and labels. We have taken a similar approach to the methods employing an EM algorithm, although we do not present our models in a probabilistic way, by proposing a mutually reinforced computation of worker confidences and aggregated labels, which is flexible and can incorporate various information. Furthermore, we provide provisions for using soft or hard labels, and discriminating or not between the quality of positive and negative labels when assessing the worker and label qualitiy.

A number of methods that do not leverage an EM approach for evaluating worker expertise and finding the hidden labels have also been proposed in the literature. Most of them rely on generative probablisitical models and Bayesian inference. We shortly review some of the proposed methods hereafter.

[WBPB10] proposes a Bayesian generative probabilistic model for the annotation process. The model infers not only the underlying class of the item, but also parameters such as item difficulty and annotator competence and bias. Furthermore, the model represents both the items and the annotators as multidimensional entities, with different high level attributes and weights. Each item has different characteristics that are represented in an abstract Euclidean space. Each annotator is modeled as a multidimensional entity with variables representing competence, expertise and bias. This allows the model to discover and represent groups of annotators that have different sets of skills and knowledge, as well as groups of items that differ qualitatively. The probability of label assignments is maximized by unsupervised MAP estimation on the parameters, performing alternating optimization on the item and worker-specific parameters using gradient ascent. The model generalizes GLAD [WWB+09] by introducing a high-dimensional concept of item difficulty and combining it with a broader definition of annotator competence. Modeling the labeling process to include label uncertainty, as well a multi-dimensional measure of the annotators' ability, as

in [WBPB10], [WP10] derives an online algorithm that estimates the most likely value of the labels and the annotator abilities. The factorized form of the general model allows for an online implementation of the EM-algorithm. Instead of asking for a fixed number of labels per item, the online algorithm actively asks for labels only for items where the target value is still uncertain.

Extending the work from [KVGHG10], which presents a family of Bayesian models for jointly learning the trustworthiness of users and truth values for statements in the presence of disagreeing user opinions and logical deduction rules, CoBayes [KGSG11] exploits user feedback and logical deduction rules in a Bayesian corroboration process. The joint inference mechanism learns the latent affinity between worker expertise and statements by taking worker and statement features into account, and mapping them into a common latent knowledge space, where the inner product between worker and statement vectors determines the probability that the worker assessment for a statement will be correct. Then, Bayesian inference is performed using mixed variational and expectation propagation message passing, and logical deduction rules are employed to interconnect assessed statements and propagate the truth values, thus mitigating feedback sparsity. Also employing a joint corroboration process [GAMS10] presents three probabilistic fix-point algorithms for aggregating disagreeing views about statements and learning their truth values as well as the trust in the views.

DARE(Difficulty-Ability-REsponse estimation model) [BGMG12] is a probabilistic graphical model that jointly models the difficulties of questions, the abilities of participants and the correct answers to questions in aptitude testing and crowdsourcing settings. By dynamically choosing the next question to be asked based on the previous responses, an active learning scheme based on a greedy minimization of expected model entropy allows for efficient resource allocation. In CrowdSynth [KHH12], a set of Bayesian models are trained for predicting the correct labels and modeling the workers and predicting their votes. These models allow the system to maintain a cost-accuracy trade-off under budget constraints by deciding whether to hire a new worker or not. [ZBMP12] considered a separate probabilistic distribution for each worker-item pair and proposed a minimax entropy principle to jointly infer the worker quality and the true labels. They argued that labels are generated by a probability distribution over workers and by maximising the entropy of this distribution the workers' quality can be naturally inferred.

A common issue when dealing with multiple, redundant judgments from workers is to aggregate them via methods like majority voting or Expectation Maximization to produce consensus labels. Unfortunately, the collected judgments are typically sparse and imbalanced, for two reasons: the average crowd worker judges few examples, and few labels are typically collected per example to reduce cost. Therefore, the consensus judgment for each example determined by only a handful of workers. While Majority Voting is completely susceptible to this problem, EM addresses this indirectly; while only workers labeling an example vote on it, global judgments are used to infer class priors and worker confusion matrices. To address this missing data

problem, [JL12a] proposes the use of probabilistic matrix factorization, to induce a latent feature vector for each person and example in order to infer unobserved judgments for all examples. Inference yields a complete matrix, which can then be used use for label aggregation. This complete matrix contains relevance judgments from all workers corresponding to all examples, and thereby reduces the bias of output consensus labels. [JL12b] further develops this approach, pointing out that once complete worker judgments are inferred, they might be used for a variety of other purposes, such as better routing or recommending appropriate tasks to workers. Following the same line of research, [Jun14] propose methods based on matrix factorization to evaluate workers and to route crowdsourced tasks to the most appropriate worker. Also addressing this issue, [VGK+14] proposes community-based Bayesian label aggregation model, which assumes that crowd workers conform to a few different types, where each type represents a group of workers with similar confusion matrices. Assuming that each worker belongs to a certain community, where the worker's confusion matrix is similar to (a perturbation of) the community's confusion matrix, they define a probabilistic Bayesian model that jointly learns latent community profiles of crowd workers, together with the individual workers' and communities' reliability profiles and the items' true labels. In a setting without repeated labels, therefore without the possibility of generating aggregate labels, [DS09] propose to simulate aggregate labels by training a hypothesis on the entire unfiltered dataset and regarding the predictions of this hypothesis as the approximate ground-truth. Intuitively, fitting a hypothesis to the entire dataset is similar to aggregating multiple labels per example. Thus, the labeler quality is estimated from the handful of provided labels, in order to prune away the low quality workers.

In a system where workers perform object comparison tasks, [VGM12] compares two quality assurance strategies: error masking and detection of bad workers using different scoring functions, and evaluate the impact on task accuracy, the number of completed microtasks, and on the cost/benefit ratio. With masking, the same task is performed by multiple workers, and some type of voting is used to select the final output. With detection, the system tries to identify bad workers and somehow discount their results. For detection, two common approaches were employed using gold standard tasks, versus plurality agreement. Also, by considering item ordering tasks, [MBKK13] proposes a statistical quality control method based on a probabilistic generative model of crowd answers by extending a distance-based order model to incorporate worker ability.

With an application to cell tower localisation, [VRJ13] addresses the problem of fusing untrustworthy reports provided from a crowd of observers, while simultaneously learning the trustworthiness of individuals. The authors construct a likelihood model of the users's trustworthiness by scaling the uncertainty of its multiple estimates with trustworthiness parameters. Then, the trust model is incorporated into a fusion method that merges estimates based on the trust parameters and an inference algorithm jointly computes the fused output and the individual trustworthiness of

the users based on a maximum likelihood framework.

[BK13] proposed an unsupervised statistical method to estimate the quality of the artifacts for a general crowdsourcing tasks with unstructured response formats. The proposed method leverages a two-stage generative model, consisting of a creation stage followed by a review stage. The creation stage models a generative process of the true artifact quality, where both the ability and the task-dependent performance of an author affect the quality of an artifact. The review stage models the generative process of the grade labels given by reviewers, where each reviewer first determines a latent quality score for a given artifact based on their bias and contextual preference, and then the observed grade label is generated through the graded response model used in the item response theory. [TL11] investigate the annotation cost vs. consensus accuracy benefit from increasing the amount of expert supervision. To maximize benefit from supervision, a semi-supervised Naïve Bayes approach which infers consensus labels using both labeled and unlabeled examples is proposed, showing that a very modest amount of supervision can provide significant benefit. To incentivize high quality outcomes in crowdsourcing, mechanisms where workers are modeled as strategic agents have also been proposed, where the participation in the task [GM12] or the proficiency of workers [DG13] are endogenous. Relying on the crowd, [ZCB11] get high quality translations in aggregate by soliciting multiple translations, redundantly editing them, and then selecting the best of them, by using a machine learning approach that assigns a score to each translation based on a set of features for both translators and translations. In an effort to control the quality of crowdsourced labeling tasks, [AMN13] shows that the reliability of workers when solving a hard task can not be assessed by combining it with a control task (that is easier to solve).

SQUARE(Statistical QUality Assurance Robustness Evaluation) [SL13a, SL13b] is an open source shared task framework including benchmark datasets, defined tasks, standard metrics, and reference implementations with empirical results for several popular methods for statistical consensus methods, based purely on the worker behaviors and latent example properties. The methods included in the benchmark are : MV (majority voting), ZC [DDCM12], DS [DS79b], GLAD [WWB+09], RY [RYZ+10] and CUBAM [WBPB10]. In addition to measuring performance on a variety of public, real crowd datasets, the benchmark also varies supervision and noise by manipulating training size and labeling error. Comparing the included methods on a diverse selection of datasets and different degrees of supervision, the authors conclude that the results vary according to the task. In our investigation we have drawn similar conclusions, that the performance of various methods depend on the task at hand.

We are proposing to use the crowd for improving machine learning, when collecting ground truth for training a supervised model, either by simply requesting labels, or by using an active learning methodology for the label gathering. We tackle the problems that are caused by the noisy nature and the questionable reliability of crowd generated answers, and in the absence of a ground truth we propose an unsupervised model to assess worker reliability and identify the hidden labels by aggregating the multiple

crowd provided labels.

Active Learning

Active Learning [CAL94] focuses on the costly acquisition of labels the instances to be used for training in machine learning. [Set10] presents a survey of Active Learning and its applications. The key idea behind active learning is that a machine learning algorithm can achieve greater accuracy with fewer training labels if it is allowed to choose the data from which it learns. An active learner may pose *queries* of unlabeled data instances to be labeled by an *oracle*. Active learning algorithms are generally evaluated by constructing learning curves, which plot the evaluation measure of interest as a function of the number of new instance queries that are labeled and added to the labeled dataset used for training. [Set10] identifies main scenarios for active learning categorized by how the learner is able to ask queries: membership query synthesis, stream-based selective sampling, and pool based sampling. Based on how they evaluate the informativeness of unlabeled instances [Set10] categorizes query strategies into: uncertainty sampling, query-by-committee, expected model change, expected error reduction, variance reduction, and density-weighted methods.

Query selection strategies have been recently surveyed in [FZL13]. As the theme of active learning is to select the most informative instances for the current model, a query strategy is employed in order to calculate instance utility based on the model prediction result represented by output probability distributions over all possible class labels. Therefore, an active labeler utilizes evaluation metrics to measure instance utility and further selects instances with maximal utility values for labeling. There are two important concepts used in utility metrics: uncertainty and correlation. By uncovering the pairwise correlation in the instance set, two instances with a large correlation value are considered similar to each other, while the two with a small value are different. Therefore, the most representative instances in a set have the largest correlations. The selected instances should form an optimal candidate set by balancing the uncertainty and diversity. On the one hand considering too much *uncertainty*, redundant instances might be selected. On the other hand when considering too much *diversity*, many uncertain instances critical for forming the boundary might be lost.

Using active learning for multimedia annotation was surveyed in in [WH11], and for natural language processing in [Ols09]. [AP11] identifies and outlines the difficulties that usually occur when employing active learning in practice.

The survey [Set10] also identifies key problems when active learning is based done with noisy oracles, such as we are employing via crowdsourcing. Some of the questions raised are: how can active learners deal with noisy oracles whose quality varies over time, how might the effect of payment influence annotation quality, and what if some instances are inherently noisy regardless of which oracle is used and repeated labeling is not likely to improve matters.

Active Learning and Crowdsourcing

Active learning traditionally considers experts as the label providers for the queried instances. Therefore the expert is assumed to be accurate, indefatigable , unique, and insensitive to costs. However, employing experts for training a model, even in an active learning setting is still costly and expensive in many cases. To reduce labeling cost, crowdsourcing the acquisition of labels, by delegating the task of labeling to inexpensive but noisy labelers, has been lately considered as a cost-effective way for active learning. Thus, by combining active learning with crowdsourcing, machine learning algorithms can be adaptively trained at a reduced cost.

Unfortunately, a direct application of crowdsourced label acquisition on active learning poses new problems. The two main plights of crowdsourcing active learning can are summarized in [FZL13]: (1) Since only a small subset of critical instances are selected for labeling, the labeling quality in active learning is more sensitive to the model's performance. (2) Since active learning is consisted of multiple learning iterations, the errors induced in each round will be passed onto the following rounds and will be amplified. Thus, asking crowdsourcing labelers to directly provide noisy class labels may not be appropriate in active learning. Therefore the main challenge of combining active learning and crowdsourcing is getting an optimal trade-off between the labeling noise and labeling cost.

To address the question about how to use noisy oracles in active learning, the learner could decide whether to query for the label of a new unlabeled instance, versus querying for repeated labels to reduce the noise of an existing training instance. [SPI08] study this problem using several heuristics that take into account estimates of both oracle and model uncertainty, and show that data can be improved by selective repeated labeling. However, their analysis assumes that all oracles are equally and consistently noisy, and annotation is a noisy process over some underlying true label. As an alternative to getting more labels for the instances, [AP10] suggests that in extreme class imbalance the labelers would be more useful if they would search for the instances of the missing class. The authors show that under extreme skew, even basic techniques for guided learning completely dominate active strategies for applying human resources to select cases for labeling, demonstrating that in such scenarios it is critical to consider the relative cost of search versus labeling. Analogous to guided learning [AP10], guided feature labeling [AMP10] provides an alternative strategy for taking advantage of human resources for improving improving supervised learning: in combination with a dual-supervision system, which allows class-polarity information about features to be taken as input, guided feature labeling allows humans quickly to prime the modeling procedure based on their background knowledge of the domain. Building on [SPI08], [ZSS11] proposes a modified active learning paradigm that considers the distribution and local consistency of labels, and aims to automatically identify those unlabeled instances that are most valuable to the learner, but also those samples that might benefit from relabeling because their initial labels seem suspect. The problematic samples are then withheld from the training

set, and thus, through the judicious use of inconsistency detection and incremental relabeling, the ability of active learning to exploit crowdsourced data is increased. Moreover, [FLZZ11] proposes an active learning paradigm, in which a nonexpert labeler is only asked whether a pair of instances belong to the same class; a MinCut algorithm is used as the base classifier and the unlabeled edge weights are repeatedly updated on the max-flow paths in the graph, in order to select those nodes with the highest prediction confidence for label acquisition. Considering the problem of classifying sentiment in political blog snippets, [HMS09] conduct an empirical study to examine the effect of noisy annotations on the performance of sentiment classification models, and evaluate the utility of annotation selection on accuracy and efficiency. They studied three dimensions: the different noise levels of annotators, the inherent ambiguity of some instances, and the informativeness of an example for the current model, concluding that good active learning or online learning schemes should take all of them into consideration.

Besides the relabeling strategy, labeling costs could also be taken into consideration. Integrating a cost budget into instance-selection metrics, guarantees the selected optimal subset subject to budget constraint. A budgeted selection task is formulated as a continuous optimization problem in [VJG10] where the optimal selected subset maximizes the improvement to the classifier's objective, with a labeling cost budget constraint. Proactive learning [DC08] focuses on selecting an optimal oracle as well as an optimal instance at the same time using a decision theoretic approach, and casting the problem as a utility optimization problem subject to a budget constraint. Emphasizing the importance of selecting both the optimal instances to be queried and the optimal oracle, the authors focus on three scenarios: oracles reluctant to give answers, oracles that charge non-uniform costs, and fallible oracles. By allowing annotators to have different noise levels, [DCS09] show that both true instance labels and individual oracle qualities can be estimated (so long as they do not change over time). They take advantage of these estimates by querying only the more reliable annotators in subsequent iterations active learning. While [DC08] provides a decision-theoretic framework to make the optimal instance-expert selection, [DCS09] generalizes from two to multiple experts, eliminating the need that one expert be a perfect oracle, and eliminating the need for explicit and reliable self-reporting of labeling confidence levels, by employing a thresholding mechanism to eliminate less reliable labelers as early as possible and boost performance. Based on the expected accuracy of labelers at each time step [DCS10] decides which annotators should be queried for a label at the next time step. Active learning with annotation time regarded as a cost is studied in [SCF08] by analysing the variability and predictability of annotation times depending on the task at hand. Tackling the multiple expert active learning scenario, [WSBT11] develop an algorithm for instance allocation that exploits the meta-cognitive abilities of novice (cheap) experts in order to make the best use of the experienced (expensive) annotators. The *difficult* label is introduced, and when an annotator employs it, it means the instance should be passed on to

someone with more expertise, assuming that novices would restrain from giving a label when they have low confidence in their ability to provide a correct label.

Active learning and crowdsourcing have been shown to be effective when used together for to enable automatic translation for low-resource language pairs in the Active Crowd Translation [AVC10] framework. Noticing that random query selection baselines are strong, as they tend to simulate the underlying data distribution when sampled in large numbers, the authors also employ a representativeness strategy for the selection of instances to be labeled. To ensure quality of translation output, each translation is requested from multiple workers, and inter-annotator agreement is used as a metric to compute translation reliability. [AHVC11] proposes active learning with multiple annotations for building comparable corpora in low-resource scenarios by requesting two kinds of annotations: class labels (for identifying comparable vs. non-parallel data) and clean parallel segments within the comparable sentences, and propose a joint selection strategy that selects a single instance beneficial for both annotations.

Besides identifying the general problems of using noisy oracles for active learning, in the literature there are also concrete solutions for coupling active learning with corwdsourcing.

In one of the first works recognizing the complementarity of active learning and crowdsourcing in lowering the costs of training sets creation, [LSS11] evaluates the trade-off between asking for more labels and labeling more examples in active learning using Amazon Mechanical Turk. Bearing a fixed amount of monetary allocations, an adaptive voting scheme is used to decide if more labels are needed for an item, or new items should be labeled. By employing only the crowd and an algorithm for posing questions efficiently [TLB+11] builds a similarity matrix between objects, by adaptively learning their embedding into an Euclidean space, referred to as the crowd kernel. The algorithm for label acquisition samples responses to adaptively chosen triplet-based relative-similarity queries, in order to maximize the information gain given previous responses. [YFRD11] addresses the question of what data to label and which crowd annotator to use at the same time, by employing a probabilistic model for learning from multiple annotators that can also learn the annotator expertise even when their expertise may not be consistently accurate across the task domain. The proposed optimization formulation allows them to select the most uncertain sample and the best worker to query the labels from for active learning. Unfortunately in systems like MTurk, worker selection mechanisms are not yet available.

An an active learning approach in which worker performance, task difficulty, and annotation reliability are jointly estimated and used to compute the risk function guiding the sample selection procedure is presented in [ZZS14]. The authors base their work on GLAD [WWB+09], which uses a probabilistic model to simultaneously estimate the labels, the labeler expertise, and the task difficulty which are represented as latent variables in the Bayesian network. The model can estimate the label of a new task with a weighted combination of labels from different labelers based on their

expertise inferred in the training phase. The proposed sampling strategy iteratively selects the combination of worker and task which offers the greatest risk reduction between the current labeling risk and the expected posterior risk, focusing on sampling reliable labelers and uncertain tasks to train a Bayesian network. Also implementing active learning in Bayesian networks with a simple structure, [TK01] use Kullback-Leibler divergence as the loss function to measure the distance between distributions. The algorithm iteratively computes the expected change in risk and makes the sample query with the greatest expected change. This strategy is guaranteed to request the label of the sample that reduces the expected risk the most, but does not account for worker performance.

Crowdsourced databases can also benefit from active learning, as shown in [MSF+12]. The authors combine the current learner performance evaluation with an unbiased estimate of its uncertainty in order to select instances for label acquisition. The selection strategies are based on bootstrap, which they use to estimate the benefit of having the crowd label an unlabeled data point. When acquiring multiple crowd labels, instead of applying the same degree of redundancy to all items, they employ a partitioning-based allocation algorithm, which partitions the unlabeled items based on their degree of difficulty for the crowd.

In the domain of computer vision, [VG14] combines active learning with crowdsourcing to train object detectors. The proposed active learning loop consists of using the current classifier to generate candidate jumping windows, storing all candidates in a hash table, querying the hash table using the hyperplane classifier, giving the actively selected examples to online annotators, taking their responses as new ground truth labeled data, and updating the classifier. By replacing the human oracle with the user tagged images obtained from social networks in [CNKK14] propose a sample selection strategy that maximizes not only the informativeness of the selected samples but also the oracle's confidence about their actual content by quantifying the samples' informativeness as the distance from the separating hyperplane of the visual model, while the oracle's confidence is measured based on the prediction of a textual classifier trained on a set of descriptors extracted using a typical bag of words approach.

We propose a framework for employing active learning and delegate the task of labeling to Crowdsourcing, motivated by the cost efficiency. Furthermore, we examine how the quality of the multiple noisy labels and different selection strategies for new instances, affect the performance of the learner.

Duplicate Detection, Crowdsourcing and Active Learning

As we test our proposed framework of actively learning from the crowd with respect to the task of finding duplicate records in a publications database, we promulgate some related work about duplicate detection. Duplicate detection has been referred to with different sobriquets: entity linkage [INN08] merge-purge [HS98], data matching [BMC+03, DLLH03], deduplication [SB02], entity identification [MVB08], refer-

ence reconciliation [HDM+05], or resolution [BGMM+09]. It describes the process of finding similar records - entity descriptions - and deciding if two descriptions refer to the same real world entity. When training a classifier for entity matching one common problem is that the data is highly skewed because the number of duplicates is much smaller than the number of non-duplicates. To address this problem [AGK10] and [BIPR12] provide active learning algorithms to maximize recall under a precision constraint.

ALIAS [SB02, SBKM02] is one of the first systems that considered leveraging active learning for deduplication. The system starts with small subsets of pairs of records designed for training a preliminary classifier. This classifier is then used for predicting the status of unlabeled pairs of records. Based on the predictions, the system iteratively seeks in the unlabeled data pool those instances which, when labeled, will improve the accuracy of the classifier at the fastest rate. ALIAS is also tested on the task of deduplication of scientific publications. [HGH+12] investigates methods for generating large-scale ground truth datasets for the deduplication of bibliographic records. The authors found that selecting duplicates and non duplicates from documents with similar titles produced more challenging datasets. Moreover, they introduce a large scale deduplication ground truth dataset based on Mendeley[2] and a Solr[3]-based technique relying on title searches.

CrowdER [WKFF12] is a system employing a hybrid human-machine approach, that uses machine-based techniques to weed out obvious non-duplicates, while using precious human resources to examine just those cases where human insight is needed. In order to get human input, the cost effective solution of using crowdsourcing is employed together methods to generate the minimum amount of HITS for a given dataset for which duplicates are to be identified. ZenCrowd [DDCM12, DDCM13] is a system for entity linking, based on a probabilistic framework leveraging both automatic techniques and human intelligence, for automatic entity extraction, ranking, and matching. ZenCrowd basically employs algorithmic matching techniques to link entities, but attempts to improve the automatic results by involving crowd workers. It resorts to human computation by dynamically generating crowdsourcing tasks in case the algorithmic components fail to come up with convincing results. Results from inverted indices, a graph database and from the crowd are using a probabilistic framework in order to make sensible decisions about candidate matches and to identify unreliable human workers.

Our framework combines active learning with crowdsourcing for training an automatic method to perform a task. We tackle the challenges that appear when employing the crowd for label acquisition in order to maintain high levels of quality, and efficiently train the automatic method. To showcase the capability of the framework through an analogous task, we demonstrate that it be applied the particular task of finding duplicate entries in a collection of publications.

[2] http://www.mendeley.com
[3] http://lucene.apache.org/solr

6

Aggregation of Crowdsourced Labels Based on Worker History

Crowdsourcing has recently gained ground as a method for gathering training data for supervised machine learning methods. As already emphasised in Chapter 5, even though crowdsourcing is more cost-effective and faster than employing experts, these advantages come at the cost of a lower label quality. While this is not pivotal in relatively simple tasks, this can be critical when dealing with complex tasks, where the labeling process requires special skills or high qualification. Different strategies have been proposed in the literature for dealing with the quality issue for crowdsourced data. One of them is to employ redundancy [SPI08], by asking different assessors to label the same item. In order to find the final label that will be used in training the machine learning algorithm, the multiple labels are aggregated according to various strategies. Majority Voting is the simplest and most widely used method for aggregating multiple crowdsourced labels, providing a satisfactory performance in most cases. This strategy assumes that workers of equal expertise cast each a vote for one label of an item. The label with most votes will finally be used. Nevertheless, this strategy has some drawbacks that can be improved upon, such as the assumption that all workers are equal and that all tasks have the same difficulty. Another main disadvantage is the unknown outcome when the voting leads to a tie. Moreover, it is not always the case that all workers have a similar level of expertise, or that that expertise level is constant during the labeling activity.

We introduce a novel yet simple model for the aggregation of multiple labels provided by the crowd. The model involves the expertise of the labeler (called *worker* in crowdsourcing scenarios) in the aggregation of the votes, by assigning a proportional weight to the votes provided for a certain item label. Hereafter, we will refer to the worker expertise as *worker confidence*. The confidence shows how much we should trust the labels provided by the worker, and what weight his labels should have in the aggregated final label. The integration of worker confidence in the aggregated label, can be done in various ways, to put more or less weight on the votes depending on

the confidence. The aggregated label can be a hard nominal binary label, or a soft numerical(e.g. real value) label indicating how confident and experienced the workers that contributed to it are, and how much they agree. We do not only model workers as having different expertise, therefore leading to us confide in them more or less, but we also have an indication of how strong the aggregated crowd label is, by means of the soft label. In assessing the worker confidence we can directly use the hard label, or we can consider the soft value, thus taking into account the strength of the aggregation process. We also investigate if considering that positive and negative answers have a different quality affects our method. Furthermore, our proposed method has special provisions for the case where self reported worker familiarity with the task at hand is available.

Our approach infers labels from multiple and possibly noisy crowdsourced labels, by applying an EM method to estimate both the workers' expertise and the actual labels. Similarly with [DS79b], the error-rate of workers, here expressed as worker confidence, is used as the weight of a worker contribution in the aggregated label computation. We propose a flexible method of computing the worker confidence, by using various ways of including additional information as well as proposing novel ways of computing the aggregated crowd label. Moreover, we introduce the soft evaluation of the worker confidence, where the soft aggregated crowd label is taken into account instead of the hard aggregated label. We provide an in-depth analysis of the proposed methods, backed by comprehensive experiments to support their efficiency. Furthermore, we also evaluate our method against state-of-the-art methods on different datasets, proving its effectiveness.

6.1 Approach

In this section we describe our flexible model for simultaneously evaluating worker confidence and crowd aggregated labels. The computation of the aggregated decision of the crowd for the label of an instance and of the evaluation of worker confidence are mutually reinforcing and will undergo a series of EM iterations until convergence or until a certain number of iterations is reached. Thus, the aggregated label depends on the confidences of the workers that contribute a label to the item, and the worker confidence depends on how the worker agrees with the aggregated labels of those items.

Let us refer to the items for which we use the crowd to get labels as i. A worker w can contribute a binary nominal label $L_w^i \in \{Yes, No\}$ to the item i. The aggregated crowd label, computed by aggregating the individual worker labels L_w^i for item i will be denoted as $L_{crowd}^i \in \{Yes, No\}$. Furthermore, let us use the boolean function $I(x)$

$$I(x) = \begin{cases} 0, & x = false \\ 1, & x = true \end{cases}$$

Each worker's expertise is characterized by his respective worker confidence, $C_w \in [0, 1]$. A confidence $C_w = 1$ would characterize a worker with perfect expertise that always gives the correct label, while a confidence of $C_w = 0$ would characterize a worker that always gives the wrong answer, thus having no usable expertise. We distinguish between two types of worker confidence depending on whether we make a discrimination between the quality of the positive and negative answers or not. The discrimination implies that a worker might manifest a different expertise when assigning positive labels than when assigning negative labels. On the contrary, without this discrimination we consider that a worker performs equally well in recognizing negative and positive examples. In the case of such a discrimination each worker is characterized by a positive confidence C_w^+ and a negative confidence C_w^-, otherwise we use a single value for the worker confidence C_w^*. Majority Voting corresponds to treating all the workers as being equally competent; in this case all the workers will be considered as being equal and having a perfect expertise, $C_w = 1$.

The process of computing the mutually reinforced L_{crowd} and C_w consists of two EM steps. In the E step we compute the aggregated crowd labels, and in the M step we update the worker confidences.

6.1.1 Aggregated Crowd Labels

The computation of the aggregated label depends on the expertise of the workers that provided labels for it. Each vote will be weighted by the confidence we have in the worker that provided it. Let us introduce the notion of **crowd aggregated soft label** for an item, $l_i^+ \in [0, 1]$ and $l_i^- \in [0, 1]$, representing the positive soft label, and the negative one respectively. The crowd aggregated soft labels, positive and negative, are an indicator of the stregth of the crowd label. When the agreement between labelers is high, or when workers with high confidence values contribute to the label, the soft label will also have a high value. The negative crowd soft label, and the strength of a negative label depends on the stregth of the positive label, and can be defined similarly to probabilities: $l_i^- = 1 - l_i^+$.

The **crowd aggregated hard label** assigned as the final binary nominal crowd label is given by comparing the positive soft aggregated crowd label, indicating how likely it is that the crowd thinks that the item should have a positive label (l_i^+) and the negative one (l_i^-). Basically if the soft positive label exceeds the value of the soft negative label, then the hard aggregated label will be positive, and vice-versa.

$$L_{crowd}^i = \begin{cases} Yes, & l_i^+ - l_i^- \geq 0 \\ No, & l_i^+ - l_i^- < 0 \end{cases}$$

Depending on whether we are discriminating between positive and negative answer quality, the crowd aggregated soft positive label can be computed as follows:

- **No discrimination between the positive and negative label quality**

$$l_i^+ = \frac{\sum_w C_w^* \cdot I(L_w^i = Yes)}{\sum_w C_w^* \cdot I(L_w^i = Yes) + \sum_w C_w^* \cdot I(L_w^i = No)} \tag{6.1}$$

  Each worker contributes to the computation of the crowd aggregated soft label with his vote weighted by his worker confidence.

- **Discrimination between the positive and negative label quality**

$$l_i^+ = \frac{\sum_w C_w^+ \cdot I(L_w^i = Yes)}{\sum_w C_w^+ \cdot I(L_w^i = Yes) + \sum_w C_w^- \cdot I(L_w^i = No)} \tag{6.2}$$

  Each worker contributes to the computation of the crowd aggregated soft label with his vote weighted by his positive confidence if he gave a positive label vote, and with a vote weighted by his negative confidence if he gave a negative label vote.

6.1.2 Worker Confidence Computation

The worker confidence is an indicator of the worker expertise and shows us how well he agrees with the labels that are assigned by the crowd as a whole. Therefore, its computation is dependent on the computation of the aggregated labels. Here too we can employ different ways for computing the worker confidence, depending on whether or not we think that the quality of negative answers differs from the quality of the positive answers. Therefore we compute the different worker confidences as follows:

- **No discrimination between the positive and negative label quality**

$$C_w^* = \frac{tp_w + tn_w}{tp_w + tn_w + fp_w + fn_w} \tag{6.3}$$

  The worker confidence is practically his accuracy, when compared to the aggregated crowd labels.

- **Discrimination between the positive and negative label quality**

$$C_w^+ = \frac{tp_w}{tp_w + fp_w} \tag{6.4}$$

$$C_w^- = \frac{tn_w}{tn_w + fn_w} \tag{6.5}$$

  The worker confidence consists of a positive confidence and a negative confidence. The positive confidence is the accuracy of the positive labels provided by the worker when compared to the crowd. Similarly, the negative confidence is the accuracy of the negative labels provided by the worker when compared to the aggregated crowd label.

Therefore, the worker confidence depends on how we compute for each worker its rate of true positives (tp_w), false positives (fp_w), true negatives (tn_w) and false negatives (fn_w) when comparing to the aggregated crowd labels. Consequently, the evaluation worker confidence can be done in two ways, depending on which type of aggregated crowd labels we use, namely:

- **hard** evaluation, where we use only the final, *crowd aggregated hard labels*,
- **soft** evaluation, where we use the *crowd aggregated soft labels*

In case of a hard evaluation of the performance of a user we use the following to compute the worker confidence in Eq. 6.3 or Eq. 6.4 and Eq. 6.5:

$$tp_w = \sum_i I(L_w^i = Yes) \cdot I(L_{crowd}^i = Yes)$$

$$tn_w = \sum_i I(L_w^i = No) \cdot I(L_{crowd}^i = No)$$

$$fp_w = \sum_i I(L_w^i = Yes) \cdot I(L_{crowd}^i = No)$$

$$fn_w = \sum_i I(L_w^i = No) \cdot I(L_{crowd}^i = Yes)$$

This is the classical way of computing the tp_w, fp_w, tn_w, and fn_w, by examining all the items for which the worker provided a label and assessing if the label coincides with the aggregated hard label provided by the crowd, depending on the type of label.

In the case of a soft evaluation of the worker confidence we use the following to compute the worker confidence in Eq. 6.3 or Eq. 6.4 and Eq. 6.5:

$$tp_w = \sum_i I(L_w^i = Yes) \cdot l_i^+$$

$$tn_w = \sum_i I(L_w^i = No) \cdot l_i^-$$

$$fp_w = \sum_i I(L_w^i = Yes) \cdot l_i^-$$

$$fn_w = \sum_i I(L_w^i = No) \cdot l_i^+$$

This involves using the crowd soft labels coupled with the answers provided by the worker, when assessing the workers confidence over all the items he provided labels for. For the true positives rate, in case the worker provided a positive answer, the positive crowd aggregated soft label of the item is taken into account. For the false positives rate, in case the worker provided a negative vote, the negative crowd aggregated soft

label of the item is taken into account. A similar reasoning holds for the true negatives rate, and the false negatives rate, respectively. This approach enables us to account for the strength of the labels, not only for their nominal assignment. Therefore, crowd aggregated labels for which the agreement is higher and the workers confidences are higher will weigh more in the evaluation of worker confidences, when compared to aggregated labels that indicate an unsure crowd decision.

The worker confidence integration in the aggregated crowd label is designed to be flexible. Thus, we can boost the confidence of the different workers when aggregating the multiple votes by using $\hat{C_w} = boost(C_w)$. The boosting function $boost(x)$ can be e^x or $x^p; p \in \mathbb{R}$. If other indicators of the worker expertise in relation to the task that he is solving are available, they can also be involved in the worker confidence, as well as in the aggregated crowd label computation. For example, the worker might provide a self-assessment of his familiarity to the task, or how good he thinks he is at solving the particular task. We also experiment with involving such a self-reported familiarity in Section 6.3.2.

6.1.3 Computation of worker confidence and crowd aggregated labels

The computation of the crowd aggregated labels as well as of the worker confidences depend on the following settings:

- how the evaluation of worker confidences is done, by using either *soft* or *hard* crowd aggregated labels (*eval*)
- the employment of positive/negative answer quality discrimination when computing the aggregated crowd label (*PN*)
- the type of boosting function applied to the worker confidence in the aggregated crowd label (*boost*)

The computation of the aggregated labels and the worker confidence are mutually reinforced. Therefore, in order to evaluate the confidence we have in the workers we can use an Expectation Maximization algorithm similarly to [DS79b]. We describe the computation in Algorithm 3. We initialize C_w , e.g. we consider all workers as equally good, with $C_w = 1$. The algorithm repeats two steps until it reaches convergence or for a certain number of iterations: (1) compute the aggregated crowd labels as decisions weighted with the worker confidences for all the items available based on the worker confidence, and (2) update the worker confidences as measures of how much the individual workers agree with the aggregated crowd label.

Algorithm 1: Worker Confidence Computation

Input: The labels of all the workers W for all the items I
Input: Method settings: Confidence Evaluation Type (soft or hard), Positive/Negative answer quality discrimination (PN), Boosting Type
Output: The worker confidences C_w and a final aggregated crowd labels L^i_{crowd} for all items

1: Initialize worker confidences with $C_w = 1$ (e.g., assume each worker is perfect)
2: **repeat**
3: Compute the aggregated soft labels L^i_{crowd} for all items using Eq. 6.2 or Eq. 6.1
4: Update all C_w using Eq. 6.3 or Eq. 6.4 and Eq. 6.5.
5: **until** confidences for all workers converged, a certain number of iterations is reached
6: **return** Workers' confidences and crowd aggregated labels for all the items

6.2 Datasets

In this section we describe the datasets which we use in the following experiments. These datasets were created for different purposes, and some of them are also recommended in the SQUARE [SL13a] benchmark for evaluating crowd consensus. They cover labeling tasks of various difficulty, gathered for diverse application domains by employing different numbers of workers and items. In Table 6.1 we present some statistics of these datasets, used to asses the performance of our proposed methods. Hereafter we give some details of the datasets.

- **HCB** [JL12a, JL11] is built from a larger dataset [TL11], containing MTurk ordinal graded relevance judgments for pairs of search queries and Web pages (i.e. not relevant, relevant, and highly-relevant), by conflating relevant classes to produce only binary labels .

- **WB** [WBPB10] contains MTurk binary judgments indicating whether images depicting 4 types of waterbirds or no bird at all, show a duck (2 of the waterbird types represent ducks) or not. Each image was annotated by 40 workers. This dataset was used to assess the author's Bayesian generative probabilistic model that performed at 75.4% accuracy, while compared with 68.3% for Majority Vote and 60.4% for GLAD.

- **WVSCM** [WWB+09] includes MTurk binary judgments distinguishing whether or not images contain smiles as either Duchenne ("enjoyment" smile) or Non-Duchenne ("social" smile). The distinction consists of the different activation of the Orbicularis Oculi muscle around the eyes. This dataset was used to show the superiority of the GLAD algorithm reporting 78.12% accuracy when compared to Majority Vote labels at 71.88%.

Table 6.1 Dataset Statistics.

Dataset	Items	Workers	Labels	GT Items
HCB	19033	762	88385	2275
WB	240	53	9600	240
WVSCM	2134	64	17729	159
RTE-RTE	800	164	8000	800
RTE-TEMP	462	76	4620	462
MEval-Label1	31076	1429	89449	5750
MEval-Label2	31039	1426	87840	5986
MMSys-Label1	4711	202	13727	13727
MMSys-Label2	4710	208	13474	13474

- **RTE** [SOJN08] includes binary judgments for textual entailment (i.e., whether one statement implies another). **RTE-TEMP** consists binary judgments for temporal ordering; the annotators assessed if one event follows another. In this dataset each example has 10 annotations. The datasets were collected with the intention of investigating how textual entailment tasks could benefit from crowdsourcing.

- **MEval** [LLBG13, LCR+14] consists of fashion-focused Creative Commons images associated with two different labels. The first label, corresponding to the **MEval-Label1** dataset, indicates whether an image is fashion-related ot not. The second label, corresponding to the **MEval-Label2**, indicates whether the fashion category of the image, represented by a clothing item, correctly characterizes the content depicted in the image. Additionally, for the second label, the workers were asked to provide their self-estimated familiarity to the fashion category. Each image is labeled by 3 different workers. A part of the images are also associated to high-fidelity labels, that can be used as a ground truth. This dataset was used in the MediaEval 2013 Crowdsourcing for Social Multimedia Task, where the complete dataset could be used for training, and the labels provided by different methods could be evaluated on the small test set having high-fidelity labels.

- **MMSys** [LMG+13] also consists of fashion-focused images collected in a similar manner to the *MEval* dataset. Therefore the **MMSys-Label1** dataset, contains labels of whether an image is fashion-related or not, and the **MMSys-Label2** contains labels of whether or not a fashion category is depicted in the image, along with the workers' self-reported familiarity to that category. Altough the dataset is similar to *MEval*, it is much smaller, but expert labels are available for all of the images, instead of just of a subset to be used for testing containg high fidelity labels.

6.3 Experiments

In this section we report the efficiency of our proposed method, first by comparing it to the Majority Voting strategy for each of the datasets, and second, by comparing the best setting for our model to other state-of-the art methods. Furthermore, we also test whether including worker self reported familiarity in the computations is providing an improvement or not. When using Majority Voting, which also coincides to the first iteration of our method (when all workers are considered to be equal), we follow the unbiased random tie-breaking strategy, without taking into consideration any class-priors. Across all experiments the stopping criterion for the EM algorithm was set to 100 iterations or convergence.

6.3.1 Performance under Different Settings

We start by investigating how our method performs with different settings on each of the available datasets. We choose the F1 measure as a performance indicator, and we compare with the Majority Voting as a baseline. In Figure 6.1 we show the F1 measure when using the different settings for our method. For the boosting function type we choose to experiment with the following 7 functions: $e^x, x^{0.5}, x^1, x^2, x^3, x^{10}$ and x^{20}. In total, in our experiments there are 28 possible settings for our methods.

For all the datasets we can notice that the increase of the boosting factor to more than x^{10} is damaging the performance. For the MEval and MMSys datasets, the boosting factor does not appear to play a deciding role, all settings providing comparable performance. This might be due to the fact that all the items are labeled just by 3 workers, and introducing a boosting factor does not affect the computation of the aggregated crowd label that much. For RTE-RTE we see that a boosting factor of x^3 provides the best performance, while applying our method on RTE-TEMP, WB and WVSCM seems not to be sensitive to this parameter. The performance on HCB seems to be affected by the choice of boosting factor, but in no clearly distinguishable way.

Whether employing a hard or soft worker confidence evaluation or the discrimination of positive and negative answer quality or not provides any benefits when compared to other settings is not clear across all the datasets. The differences in performance produced by the different settings cannot lead us to strong conclusions. Overall, most settings lead to a performance improvement when comparing to the Majority Voting strategy. For the fashion-domain datasets and for RTE-RTE, it seems that the hard and soft evaluations provide the same performance, and generally the inclusion of a discrimination between positive and negative labels hurts the performance. For RTE-TEMP, a hard worker evaluation is always better than a soft one, and the inclusion of the positive and negative label discrimination is beneficial.

In Table 6.2 we report the performance measures for the Majority Voting strategy. This constitutes the baseline for comparing the performance of the different settings

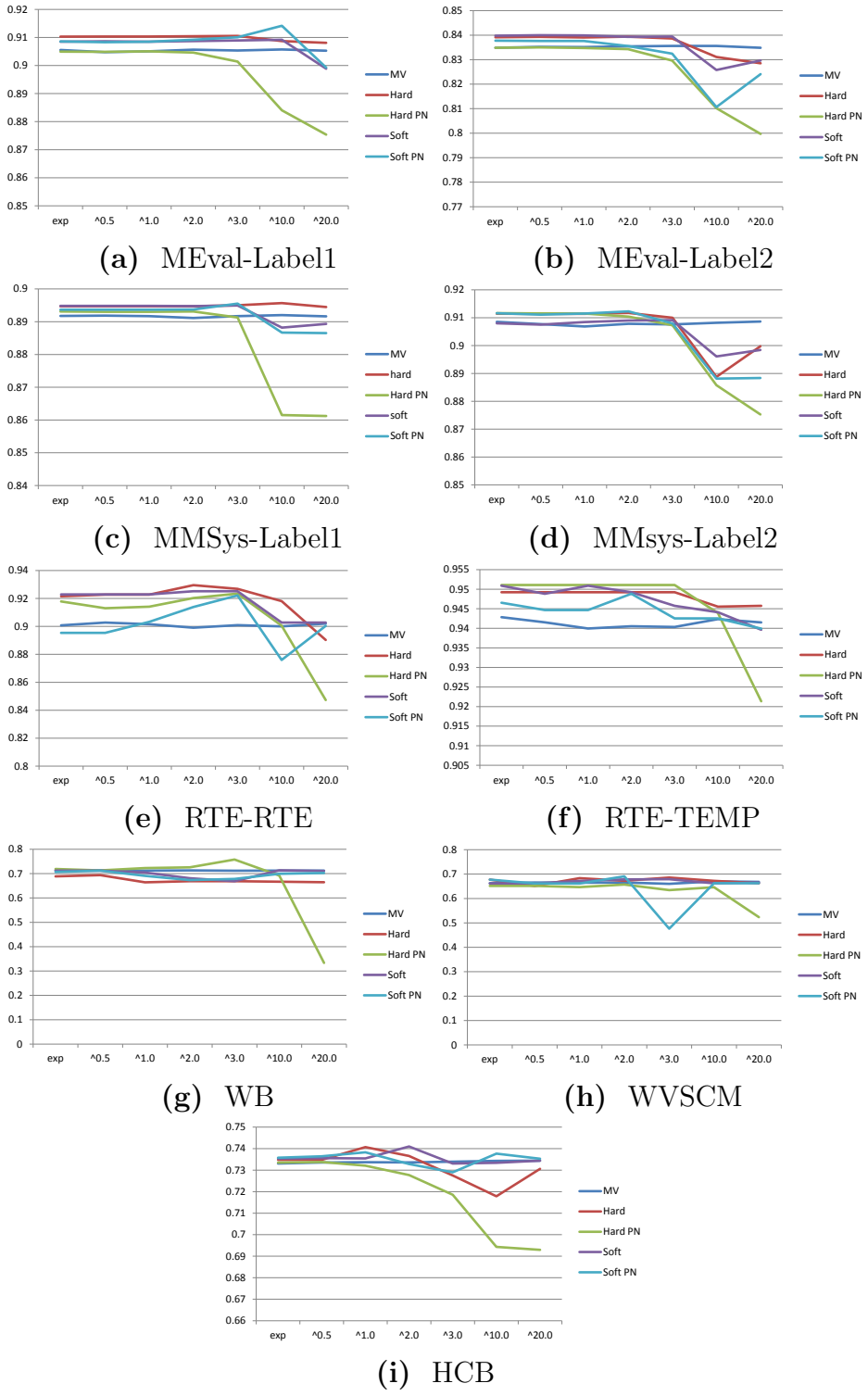


Figure 6.1 F1 Measure on the different Datasets

Table 6.2 Performance of the Majority Voting strategy on each dataset.

Dataset	Acc	F1
HCB	0.6523	0.7357
WB	0.6833	0.7099
WVSCM	0.7233	0.6667
RTE-RTE	0.8875	0.8931
RTE-TEMP	0.9437	0.9486
MEval-Label1	0.8793	0.9067
MEval-Label2	0.8657	0.8367
MMSys-Label1	0.8901	0.8906
MMSys-Label2	0.9281	0.9059

as well as of other methods. We will evaluate the effectiveness of the different methods by comparing how much they improve over using the simple yet powerful Majority Voting strategy. The performance of this strategy is dependent on the employed dataset as we can see from the Table. Nevertheless, depending on the dataset and the method used, this strategy can be a very strong baseline to beat.

In Table 6.3 we report the settings that achieved the best performance in terms of accuracy on each of the datasets. In terms of F1 we report the settings achieving best performance in Table 6.4. There is no setting that clearly provides the best results across all datasets. We can observe that while for the fashion datasets employing the soft worker evaluation provides the best results, for the other datasets, employing the classical hard worker evaluation provides better results. Compared to the performance of Majority Voting, reported in Table 6.2 we can notice that in terms of accuracy, our method is better than Majority Voting across most datasets. For HCB which contains labels that were conflated from relevance judgements, our method is not providing a higher accuracy than employing Majority Voting. In terms of F1, our method is always providing a performance increase. We are more interested in the F1 measure because we are interested in having high quality in both the positive and negative labels. Similarly to the Majority Voting, the performance of our methods oscillates depending on which dataset we apply it to.

6.3.2 Incorporating Self-Reported Familiarity

As already mentioned in Section 6.2, the requester of crowdsourced work can ask the workers to provide a self-reported familiarity to the task. In this section we propose methods for integrating the familiarity in the worker confidence, and investigate how this affects the overall performance of the proposed methods.

We test the involvement of the worker self-reported familiarity on *MEval-Label2* and *MMSys-Label2*, as they are the only datasets where it is available. The worker is here requested to indicate how familiar he is with a certain fashion item or category,

Table 6.3 Accuracy for the best settings of our method.

Dataset	Evaluation	PN	Boost	Acc
HCB	soft	✓	x^{10}	0.6462
WB	hard	✓	x^{10}	0.7958
WVSCM	hard	-	x^3	0.7925
RTE-RTE	hard	-	x^2	0.9288
RTE-TEMP	hard	-	e^x	0.9437
MEval-Label1	soft	✓	x^{10}	0.8869
MEval-Label2	soft	-	e^x	0.8685
MMSys-Label1	hard	-	x^{10}	0.8957
MMSys-Label2	soft	✓	x^2	0.9123

Table 6.4 F1 for the best settings of our method.

Dataset	Evaluation	PN	Boost	F1
HCB	soft	-	x^2	0.7410
WB	hard	✓	x^3	0.7577
WVSCM	hard	-	x^3	0.6857
RTE-RTE	hard	-	x^2	0.9295
RTE-TEMP	hard	✓	x^1	0.9511
MEval-Label1	soft	✓	x^{10}	0.9142
MEval-Label2	soft	-	$x^{0.5}$	0.8400
MMSys-Label1	soft	✓	x^3	0.8950
MMSys-Label2	soft	✓	x^2	0.9336

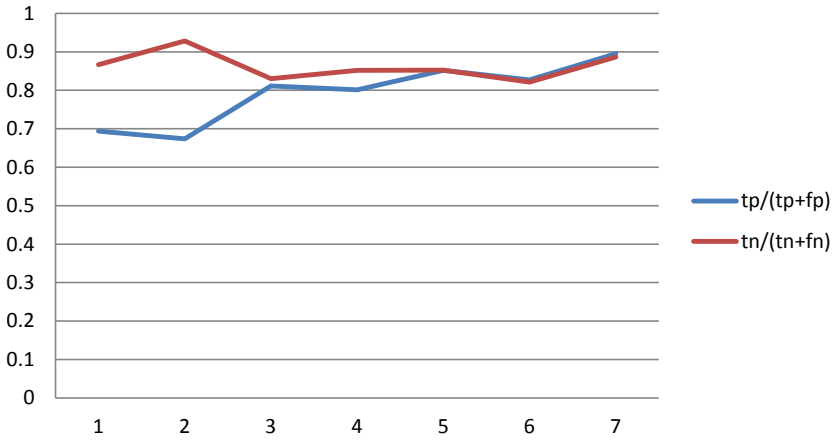


Figure 6.2 Correlation between answer accuracy and reported familiarity for MMSys-Label2

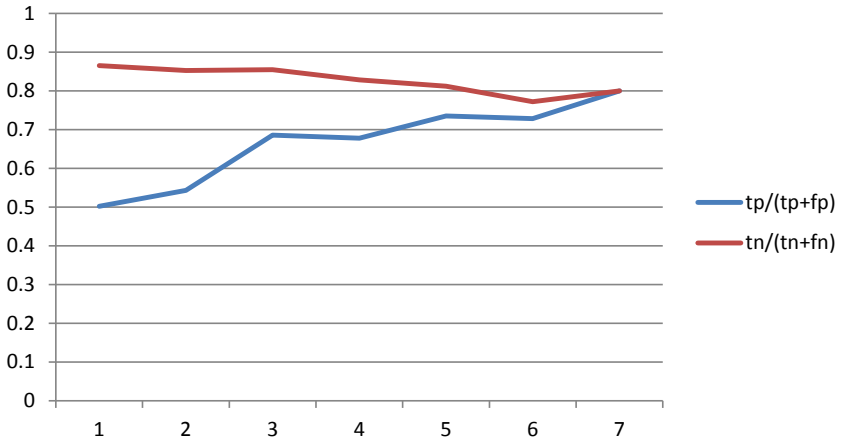


Figure 6.3 Correlation between answer accuracy and reported familiarity for MEval-Label2

by giving an integer in the range from 1 to 7, with higher values indicating a higher self-assessed familiarity. We integrate the worker self-reported familiarity to the category for which the label is assigned to the image (fam_w^i) in the computation of the confidence by using $\check{C}_w = C_w \cdot norm(fam_w^i)$. The transformation of familiarity from an integer within 1 and 7 or missing to a real subunitary positive number, is done by the $norm(x)$ function. $norm(x) = (x-1)/6$ if $x \in \mathbb{N}$ and 0.5 if missing.

In Figure 6.2 and Figure 6.3 we plot the positive answer accuracy rate ($tp/(tp+fp)$) and the negative answer accuracy rate($tn/(tn+fn)$) for each level of self-reported familiarity for the *MMSys* and *MEval* datasets respectively. We can observe a higher accuracy on giving negative answers when the self-reported familiarity is low. As the self-reported familiarity grows, the accuracy for both labels seems to stabilize and be of equal dimension. Based this observation on the correlation of the familiarity

Table 6.5 The number of cases where the performance was increased(F+, FC+) or decreased (F-, FC-), as well as the maximum performance achieved(Max F, Max FC), and the the maximum performance increase(Max-I F, Max-I FC) in terms of F1 when involving the self-reported familiarity in the worker confidence computation(F), respectively using the familiarity correction deduced from our observations(FC).

Dataset	Eval	F+	F-	Max F	Max-I F	FC+	FC-	Max FC	Max-I FC
MMEval-L2	H	6	8	0.8393	0.0336	8	6	0.8393	0.0328
MMEval-L2	S	9	5	0.8405	0.0747	10	4	0.8404	0.0749
MMSys-L2	H	3	11	0.9094	0.0193	4	10	0.9111	0.0337
MMSys-L2	S	9	5	0.9093	0.0419	10	4	0.9104	0.0521

and the type of answers and their accuracy, we can also use a familiarity correction strategy in the computation of the worker confidence.

$$\hat{C}_w = \begin{cases} 0.6 & fam_w^i < 3, L_w^i = Yes \\ 0.9 & fam_w^i < 3, L_w^i = No \\ 0.8 & fam_w^i \geq 3, L_w^i = Yes \\ 0.8 & fam_w^i \geq 3, L_w^i = No \end{cases}$$

Therefore, if we involve the self-reported familiarity in the computation of the worker confidence, the method will have two more settings:

- the involvement of familiarity in the computation of the worker confidence (F)
- the use of the familiarity to correct the worker confidence based on the observation of the correlation of the familarity with answer quality(FC)

We evaluate in how many of the possible settings for our method the addition of the familiarity to the computation provides an increase in the F1 measure. Considering all the possible settings for the method: evaluation of worker confidence(soft/hard), discrimination between positive and negative answer quality, and the boosting factor, in each setting we involve the familiarity. This amounts to 14 different settings for each choice of evaluation of worker confidence. We evaluate if the addition of one of the modalities of involving the familiarity provides a performance increase or decrease when compared to the same setting without, by counting in how many of the cases, the performance increases or decreases. We discriminate between the soft and hard evaluation and we present the results together with the the highest performance achieved and the highest increase for each general setting in Table 6.5.

We can notice that the involvement of the familiarity in the computation is more successful when employing a soft evaluation of the worker confidence. This provides an increase for most of the method settings and also provides a much higher maximum increase than in the case of a hard evaluation. Moreover, applying the familiarity

correction directly provides a higher increase than just involving the familiarity. Nevertheless, in a real life scenario, the familiarity correction has to be based on a close understanding of the worker behavior, and how the self-reported familiarity correction translates into actual work quality. In our experiments we take advantage of the fact that a ground truth is available and investigate how the correction should be made depending on the familiarity. This ground truth might not always be available, and other methods should be employed to investigate how the self-reported familiarity correlates with their expertise and performance. We can notice a performance increase in less cases when employing the hard evaluation of worker confidence, and the maximum performance increase is lower than in the case of a soft evaluation. In terms of the highest performance obtained, the differences amount to an improvement of around 0.5% over the Majority Voting for both *MEval* and for *MMSys*. Therefore, we can say that the involvement of familiarity is mostly beneficial when coupled with a soft evaluation of the worker confidences, providing a serious performance increase, on both the examined datasets.

6.3.3 Comparison to Other Methods

We compare the method settings that provided the best results for the cases where familiarity was not involved, presented in Table 6.4 and Table 6.3, to EM-based state-of the art methods in terms of both F1 and accuracy. Although we believe that the F1 measure offers more insights into the performance of such methods, we also considered accuracy because this is the measure used in the papers that introduced the state-of-the-art methods we are comparing against. We have chosen only methods that are based on a similar EM simultaneous estimation of worker reliability and item labels. Hereafter we briefly describe these methods.

1. ZenCrowd (ZC) [DDCM12] weights the votes of the workers according to their corresponding reliability, and employs an Expectation Maximization algorithm to simultaneously estimate the hidden labels and the worker reliability. ZC probabilistically models workers as acting independently, their behaviour being also independent of each item's true class assignment. The model tackles sparse data well, because of its reduced complexity.

2. Dawid-Skene(DS) [DS79a] has become the classical approach alongside Majority Voting. It models a confusion matrix for each worker and a class prior, by simultaneously estimating labels, confusion matrices and the class priors using Expectation Maximization. The confusion matrices enable modeling worker reliability as depending on each item's true class. The weakness of this algorithm is that it is easily affected by sparsity.

3. Raykar(RY) [RYZ+10] estimates the error-rates and the underlying hidden labels in the absence of a golden standard, by employing a Bayesian approach

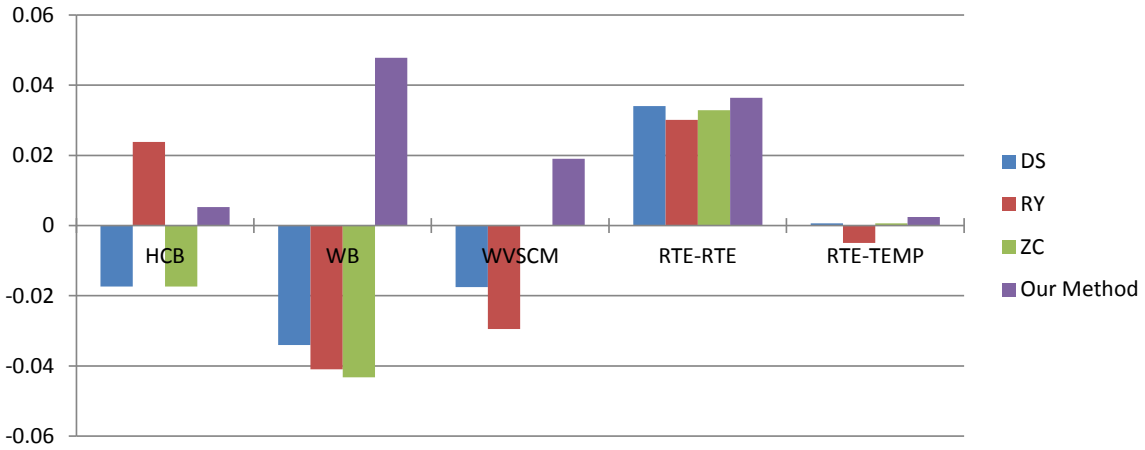


Figure 6.4 F1 difference in comparing to Majority Voting compared with other methods on the datasets on the non-fashion domain datasets

and worker priors for each class. Each worker is modeled as having a bias towards sensitivity (bias towards the positive class) or specificity (bias towards the negative class). This is similar to the discrimination we propose between positive and negative answer quality for a worker.

For DS we use the implementation provided in [IPW10], while for ZC and RY we use the SQUARE [SL13a] toolkit.

In Figure 6.4 and Figure 6.5 we present the performance difference when compared to the MV strategy in terms of the F1-Measure. For the sake of completeness we also report the accuracy difference in Figure 6.6.

Investigating Figure 6.4, containing an evaluation done on diverse datasets, close to the usual employment of crowdsourcing we can notice our method always leads to an increase in performance when compared to the Majority Voting, as already mentioned in Section 6.3.1. HCB is the only dataset where our method is not the best performer, being surpassed by Raykar. For WB and WVSCM our method is the only one that produces an increase, while on RTE-RTE our method provides the highest increase. On RTE-TEMP it also provides an increase, while Raykar and ZenCrowd produce a slight decrease in performance when compared to the Majority Voting strategy. Our method also provides better performance on WB and WCSCM, and worse on HCB.

In Figure 6.5, focused on evaluating on the fashion-specific domain datasets, we can notice that our method is outperformed by Raykar for the first label on both datasets, while our method provides the highest F1 increase for the second label. Furthermore the increase is much bigger for the smaller MMSys dataset. For the first label, whether or not an image is fashion related, which is an interpretation sensitive question, the improvement of all methods is low. On the second label, which was related to the identification of a fashion item in the photo, which is a clearer task,

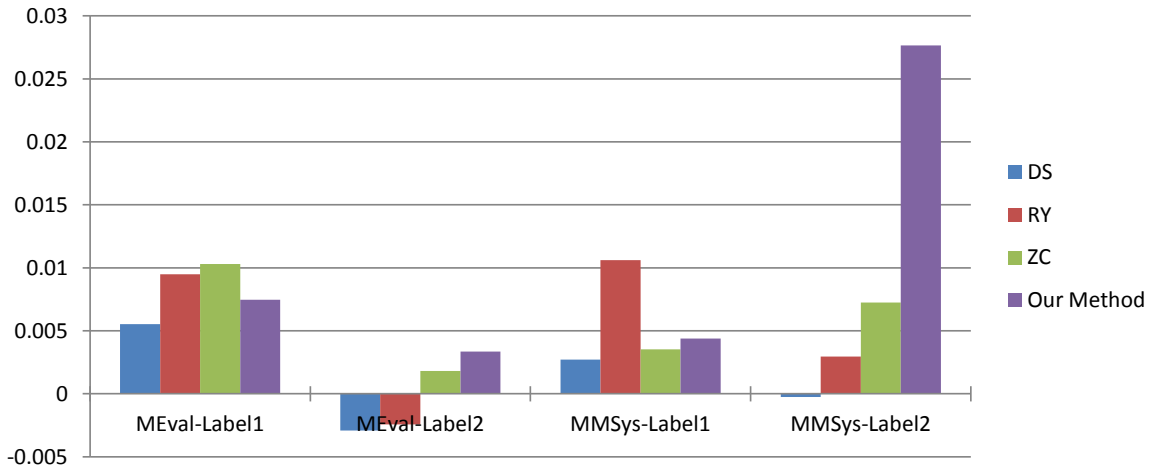


Figure 6.5 F1 difference in comparing to Majority Voting on the MEval and MMSys datasets

but requires more expertise, the performances of all the methods are close together for the larger MEval dataset, while for the MMSys dataset, our method provides the highest increase.

The non-fashion domain datasets produce higher performance differences when compared to Majority Voting and among the methods themselves. In the fashion domain datasets, the fact that the items have a number of labels limited to 3 hinders the performance of all the methods. On the contrary the other datasets , have more labels available per each item, making the modeling of the workers and the computation of the aggregated crowd label more reliable.

In terms of accuracy, as we can see from Figure 6.6, for the fashion-domain datasets there is no major difference between the performance of all the methods when compared to the Majority Voting. The same holds for the RTE-TEMP dataset. For WB and WVSCM our method has the best improvement, while most other methods lead to a decrease in performance. On RTE-RTE the all the methods lead to a similar increase in performance with no clear best performer. For HCB, our method leads to a decrease in performance, while the only method that leads to an increase is Raykar.

6.4 Conclusions

We have introduced a novel method for the aggregation of crowd labels in order to find the underlying hidden labels, while at the same time estimating the worker quality. Our model is based on an EM technique where the computation of the aggregated worker labels is reinforced by the computation of worker confidences. The model is flexible and can account for different ways of assessing the worker reliability. Our method can incorporate the hard nominal binary crowd aggregated labels in

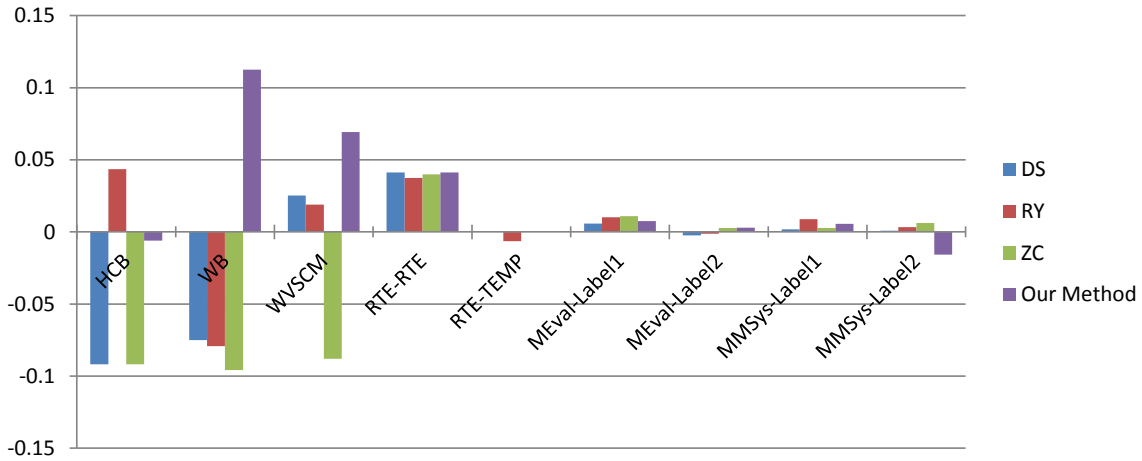


Figure 6.6 Accuracy difference to majority voting compared with other methods on all datasets

the evaluation of worker confidence, or take the soft labels, that indicate the label strength. We investigate the effect of discriminating between positive and negative answer quality, as well as different ways of boosting individual worker confidences in the aggregated label.

Through extensive experimentation on diverse datasets we have demonstrated the efficiency of our methods. When compared to state-of-the art methods, the advantage of using our method depends on the chosen dataset. While on some datasets it is showing a clear improvement over its contenders, this is the opposite on other datasets. This shows that the efficiency of any method highly depends on the environment where it is deployed, and the underlying labeling task is very important.

Future directions for this work include testing the proposed methods on synthetic data, and testing the noise resistance. Similar to other methods we can try to introduce different levels of supervision into the algorithms. Further extensions can be done to the computation of the worker confidence, by incorporating a part of the ground truth, or by simulating the behaviour of good users, based on the features of the tasks where they performed well. An automatic mechanism that refuses labels from workers that were identified as being unreliable might benefit the requester of crowdsourced work, by keeping the costs low, and gathering just higher quality labels.

7

Active Learning with Crowdsourcing

In this chapter we propose a framework for an automatic method to incrementally learn from the crowd how to perform a certain task. As emphasised in Chapter 5, the combination of active learning and crowdsourcing can be a cost-efficient way of trainig a machine learning algorithm. In our proposed framework, the labels are provided by the crowd, on demand, for instances selected according to an active learning methodology. In each round the acquired labeled instances must maximize the information gain of the learner. We have included special provisions to take into consideration the noisy nature of crowd labels, and the diversity of workers. To this end, our methods can employ crowd label aggregation schemes that consider worker expertise. Furthermore, for instances with inconclusive aggregated labels, more labels should be collected from the workers, in order to build a reliable training set for the automatic method. Each round of active learning has a corresponding cost that can be controlled with respect to the resources allocated to it. Only an optimal combination of selection strategy for new instances, allocation of resources, and crowd label aggregation will lead to a balance between cost and performance.

We identify and tackle some of the challenges raised by actively learning from the crowd. We experiment with different automatic methods, each learning from the crowd in its own distinct manner. For each round of active learning, resources are allocated to gather labels for new examples or for examples that persist with inconclusive aggregated labels. We carry out experiments to determine the optimal balance between the allocated resources and the performance increase of the trained automatic algorithm. Moreover, the instances for which new labels are gathered in each active learning round can be selected according to different strategies. We experiment with uncertainty, representativeness as well as a random selection strategy, confirming previous findings that a strategy which chooses instances that are representative of the entire pool of unlabeled data performs best. Despite the fact that the reliability of crowdsourced work is questionable, with respect to its concomitatnt low costs when compared to employing experts, it can have net advantages.

Although the proposed methods can be applied to any domain, we focus on the special case of duplicate detection. Futhermore, we focus on the particular case where the entities are scientific publications. By duplicates we mean metadata documents that refer to the same real-world publication. As an example, consider an entity described by the following metadata: *title="As We May Think", author="Vannevar Bush", year="1945", book="The Atlantic"*. In different sources the fields might be represented differently, or in a source relying on OCR there might be parsing errors or spelling errors, or some publications are present multiple times, each time with small differences in attribute values. For instance another representation of the same publication can be: *title="As we may think", author="V. Bush", journal="Atlantic Magazine"*. A publication search system needs to identify entities that match this criteria, in order simplify results lists, by grouping them at query time.

Various automatic methods can be used for computing labels for a pair of entities, using features learned by our proposed method from training data gathered from Amazon's Mechanical Turk[1] in an active learning manner. In each round we extend the available training set in order to improve the performance of the automatic method, guided by the training set or the settings learned in the previous rounds.

7.1 Framework for Active Learning from the Crowd

7.1.1 Active Learning from the Crowd

We are employing an active learning technique to improve an *Automatic Method* so that its behavior fits better to how the crowd solves a particular task. Unlike an ordinary learner that is trained using a static training set, an active learner actively picks subsets of instances from the unlabeled data which, when labeled, will provide the highest information gain to the learner.

The general steps taken by our method are described in in Algorithm 2. We start with an empty list of instances that need to be labeled (*Candidates*), an empty training set (*Train*) and an empty set of workers that need to be excluded from the crowd label assignment process (*BadWorkers*). We use a *Selection Strategy* for guiding the active learning process, and a *Label Aggregation Strategy* to aggregate the crowd labels (CSL) and in the same time assess the worker performance (WQ). Two thresholds are used in the algorithm: the worker confidence threshold *WQThreshold* and the aggregated crowd label confidence threshold *CSLThreshold*. As an output of every active learning round the algorithm provides a training set for the Automatic Algorithm, *Train*, that will get better and better with each iteration, the workers that were identified to provide labels of unsatisfactory quality, *BadWorkers*, and the instances for which we need to get more labels in order to have a conclusive and confident aggregated crowd label, *Candidates*.

[1] http://www.mturk.com

Algorithm 2: Active Learning from the Crowd.

Input: Sample size: s instances per round
$AutomaticMethod$
$LabelAggregationStrategy$
$SelectionStrategy$
$WQThreshold$; $CSLThreshold$

Output: $Train$: Training set for the $AutomaticMethod$
$BadWorkers$: Low quality workers
$Candidates$: Instances to label

1: $Candidates = \varnothing$
2: $Train = \varnothing$
3: $BadWorkers = \varnothing$
4: Use default or common sense parameters for the $AutomaticMethod$
5: **loop**
6: Add to $Candidates$ a set of instances of size s chosen by the $SelectionStrategy$
7: Prepare a batch of micro tasks containing instances in $Candidates$
8: Post micro tasks on the crowdscourcing marketplace excluding workers from $BadWorkers$
9: Retrieve labels from the crowdsourcing marketplace
10: Compute the aggregated crowd labels CL and CSL and assess the worker confidences WQ using the desired $LabelAggregationStrategy$
11: Add workers with $WQ < WQThreshold$ to $BadWorkers$
12: Add instances with $CSL \geq CSLThreshold$ to $HighConfidence$
13: $Candidates = Candidates - HighConfidence$ (Keep the instances for which the confidence in the aggregated label is not strong enough for the next iteration in order to get more crowd labels)
14: $Train = Train + HighConfidence$ (Add the high confidence instances and their aggregated labels to the training set for the Automatic Method)
15: Retrain the Automatic Method with the new $Train$
16: **return** $Train$, $BadWorkers$, $Candidates$
17: **end loop**

In consecutive rounds we repeat the same learning process: To the instances that still have an uncertain status, *Candidates*, we add a sample of instances provided by a candidate instances *Selection Strategy*. We prepare a batch of micro tasks to be solved by the crowd, blocking first the workers that were identified to be unreliable, *BadWorkers*. After the microtasks are solved by the crowd, we compute the aggregated crowd labels and the worker confidences using the desired *Label Aggregation Strategy*. We monitor the worker confidence, and the workers that fall below a certain accountability threshold can be blocked so that they do not continue to

introduce errors to our system. We identify the *High Confidence* set, the instances for which the confidence and agreement between workers indicate a clear label. We use this set to improve the performance of our Automatic Method. If we do not have a high agreement between the workers, or if the confidence in the aggregated crowd label is not high enough, then we need to get more opinions on these instances, by extending the number of assignments of the corresponding microtasks, and keeping them in the *Candidates* set. At the end of each round we extend the training set by acquiring labels that would increase the performance of the automatic method, and we identify workers that are not reliable and instances for which we need to get more labels from the crowd. We can stop the active learning process and the loop once we get a satisfactory performance using the training set built over the consecutive rounds, or we reach the limits of the specified budget.

7.1.2 Gathering Labels from the Crowd

In order to optimize the Automatic Method such that it provides results as close to reality, we use a crowdsourcing marketplace like Amazon's Mechanical Turk (MTurk), since it can provide labeled data for machine learning rapidly and at a low cost. Consequently, we delegate the task of assigning labels to MTurk workers. On MTurk a unit of work is called a *HIT*, Human Intelligence Task, and it can be done in five minutes or less, for a monetary reward. Each HIT has an extendable maximum number of *assignments*, indicating how many distinct workers should solve the same task, for quality assurance through redundancy. The data is sent to MTurk in batches. With each solved batch, the automatic method learns from the crowd how to do the task better. The automatic method thus improves; although there will always be some instances on which the label assignment is uncertain, this number will decrease, together with the need for input from the crowd.

7.1.3 Aggregation of Crowd Labels

Various *Label Aggregation Strategies* can be used for aggregating the crowd labels. The aggregation of different labels provided by the crowd workers can take into account features of the instances to be labeled, as well as worker features. Let us refer to the aggregated *Crowd Label* of an instance i as $CL(i) \in \{1, -1\}$. For the task of duplicates detection 1 stands for duplicates and -1 for non-duplicates. It can have an associated score, that is an indicator of the confidence we have in it being the true label, called *aggregated label confidence* or aggregated *Crowd Soft Label*, $CSL(i) \in [0, 1]$. The confidence we have in the label tells us if we need to get more labels for the instance, in order to strengthen the aggregated label. If the confidence surpasses a chosen threshold, *CSLThreshold*, then we do not need any more labels and we can use the aggregated label as it is in the training of the automatic algorithm. A simple measure can be the agreement between the workers providing the labels. Furthermore, the

confidence of the aggregated label can allow more advanced automatic methods to take label uncertainty into account when training.

To compute an aggregated crowd label, the difference in quality of labels provided by different workers has to be taken into account. Therefore, we have to assess the quality of the labels provided by each worker; we will further refer to this measure as worker confidence, WQ. Some workers perform better than others depending on their understanding of the task and their background and experience. The weight that a label has in the aggregation should be proportional to how good the worker providing it is. We want to penalize the labels coming from workers that do not provide good quality answers, and boost the weight of the answers of those workers that are good. Eventually, we build a database of underachieving workers, the *BadWorkers*, and block them from participating in our tasks. These are the workers with an evaluated WQ over one or multiple consecutive rounds that falls under the acceptable worker confidence threshold, *WQThreshold*. Eliminating the workers that are consistently providing low quality answers would reduce the noise and lead to better labels.

7.1.4 Computing the Aggregated Crowd Label

As already mentioned, although the methods proposed can be applied to any domain, we focus on the special case of duplicate entity detection, tackling the particular case where the entities are scientific publications.

Let us use the following notations:

e_i is an entity, described as a set of attribute-value tuples:

$$e_i = \left\{(F_k, V^i_{F_k}))|F_k \in FieldNames\right\}$$

,where F_k denotes the field name and $V^i_{F_k}$ its value for entity e_i. The field name F_k belongs to a fixed set of possible field names, $FieldNames$

$p_{i,j}$ denotes a pair of entities (e_i, e_j) that can be duplicates or not.

w_k represents a worker that can provide a label for a pair of entities

$W_{i,j} = \{w_k|w_k$ assigned a label for $p_{i,j})\}$ is the set that contains all workers that labeled the $p_{i,j}$ pair.

$WL_k(p_{i,j})$ the label that worker w_k assigned to the entities pair

$$WL_k(p_{i,j}) = \begin{cases} 1, & \text{worker } w_k \text{ assigned to } p_{i,j} \text{ a duplicates label} \\ -1, & \text{worker } w_k \text{ assigned to } p_{i,j} \text{ a non-duplicates label} \end{cases}$$

P_k is a set containing all entity pairs for which w_k provided labels.

P is the set containing all entity pairs labeled by the crowd.

From Mechanical Turk we get tuples of the form $(p_{i,j}, w_k, WL_k(p_{i,j}))$ and we want to aggregate the labels from individual workers in order to get an assignment that is as close as possible to the real case.

Let us use $CL(p_{i,j})$ as the *aggregated Crowd Label*, computed for the $p_{i,j}$ by using all the $WL_k(p_{i,j})$ of the workers in $W_{i,j}$

$$CL(p_{i,j}) = \begin{cases} 1, & \text{the crowd assigned a duplicates label to } p_{i,j} \\ -1, & \text{the crowd assigned a non-duplicates label to } p_{i,j} \end{cases}$$

We also define an *aggregated Crowd Soft Label* $CSL(p_{i,j})$ that aggregates the crowd labels into a number in the $[0, 1]$ interval. The $CSL(p_{i,j})$ also gives an indicator as to how much the worker agree, and how strong they feel about the $p_{i,j}$ pair.

Each worker w_k has an associated confidence c_k, indicating how good he is in solving the tasks, $c_k \in [0, 1]$

$weight_k(p_{i,j})$ represents the weight that the label of worker w_k has in the computation of the aggregated crowd label over the $p_{i,j}$ pair

The aggregated crowd decision for a pair is

$$CD(p_{i,j}) = \sum_{k \in W_{i,j}} weight_k(p_{i,j}) * WL_k(p_{i,j})$$

Each worker that gave a judgment about the pair will have a different weight in the decision. This weight is computed based on his confidence value. For a pair of entities the weights add up to 1.

The weight that worker w_k has in the decision for the pair $p_{i,j}$ is

$$weight_k(p_{i,j}) = \frac{c_k}{\sum_{v \in W_{i,j}} c_v} \tag{7.1}$$

We can boost the weight of the workers having high confidence over that of the low-confidence workers, by using a boosted weight

$$weight_k^{boost}(p_{i,j}) = \frac{e^{c_k}}{\sum_{v \in W_{i,j}} e^{c_v}} \tag{7.2}$$

The aggregated crowd label can therefore be computed as

$$CL(p_{i,j}) = \begin{cases} 1, & CD(p_{i,j}) \geq 0 \\ -1, & CD(p_{i,j}) < 0 \end{cases}$$

Because $CD(p_{i,j})$ can vary between -1 and 1, order to make CSL comparable to ASL we will bring it into the $[0, 1]$ interval. The *crowd soft label* is therefore defined as

$$CSL(p_{i,j}) = \frac{1 + \sum_{k \in W_{i,j}} weight_k(p_{i,j}) * WL_k(p_{i,j})}{2} \tag{7.3}$$

Using the aggregated crowd soft label, we can assign a hard label the $p_{i,j}$ pair

$$CL(p_{i,j}) = \begin{cases} 1, & CSL(p_{i,j}) \geq 0.5 \\ -1, & CSL(p_{i,j}) < 0.5 \end{cases}$$

7.1.5 Quality Control for the Crowdsourced Work

Because not all workers have the same expertise or motivation, we need to identify the good workers and rely on them, and to ignore the the bad workers, and even not allow them to participate in solving new batches, in order to reduce noise.

Testing Workers Before They Are Allowed to Solve HITs

In order to protect requesters from low quality workers Mechanical Turk offers the possibility to assign *qualifications* to workers. This can be done manually, but it can also be automated, via a *qualification test*. A qualification test consists of a number of pairs on which we know the answers. The workers are evaluated if they understood the task and if they are able to solve it. After taking the qualification test, the workers are assigned a score for the particular qualification. The requester can put a constraint on the workers, allowing only those having a qualification score greater than a specified threshold to solve the HITs. We posted a batch of 60 HITs with 3 assignments, each consisting of 5 pairs to be deduplicated, for which the workers were required to take a qualification test we devised. This proved not to be a very fruitful approach, as not many workers took our HITs. It might be that the extra work that faced them when they encountered the test, drove them away, or they did not pass it with a satisfying score. To complete this batch of tasks the workers needed more than a month. In comparison, in a comparable setting where there was no prerequisite of having passed the test, the batch was finished in a matter of 1-2 days.

Worker Confidence

A simple metric for evaluating the worker confidence can be the proportion of correctly assigned labels, when compared to the crowd aggregated labels.

$$c_k = \frac{\|\{p_{i,j} | p_{i,j} \in P_k \text{ and } WL_k(p_{i,j}) = CL(p_{i,j})\}\|}{\|P_k\|} \tag{7.4}$$

In order to evaluate the confidence we have in the workers we can use a Expectation Maximization algorithm as proposed in [DS79b], described in Algorithm 3. We initialize c_k with an initial values, e.g. all workers are considered equally good, with $c_k = 1$. The algorithm repeats two steps until it reaches convergence or for a certain number of iterations: (1) compute aggregated labels for all available pairs based on the worker confidence, and (2) update the worker confidences.

Improving the Quality of Workers

Running the learning algorithm on the output of a certain worker we could learn which are the fields that have more importance when she is doing the classification task. We could learn how workers identify duplicates, see where they do wrong and why,

Algorithm 3: Worker Confidence Computation.

Input: The labels assigned by the workers $WL_k(p_{i,j})$ for all pairs
Output: The worker confidences w_k and a final label for all pairs

1: Initialize worker confidences with $c_k = 1$ (e.g., assume each worker is perfect)
2: **repeat**
3: Compute $CSL(p_{i,j})$ for all $p_{i,j}$ pairs using Equation 7.3
4: Update all c_k using Equation 7.4
5: **until** all worker confidences converged
6: **return** Workers' confidences and aggregated crowd labels for all the pairs

and then recommend them ways to improve the quality of their work by indicating that they do not pay enough importance to certain fields, or that they ignore some fields that should also be considered. We could identify the most common mistakes and also improve the description of the micro tasks, or provide better examples that match the pairs that are most often badly classified. Therefore, returning to the original computational paradigm where the machine helps the human, the automatic algorithm can tell the human workers where they are doing mistakes, and suggest them some fixes that will improve the quality of their work. A mechanism that can offer workers a feedback from the automatic algorithm, would be beneficial for both parts, and the quality of the overall system would increase.

7.1.6 Candidate Instances Selection Strategies

In each round the instances that will get new labels from the crowd workers are selected using a *Selection Strategy*. The way these instances are selected influences how fast the Automatic Method learns. The selection strategy guides the active learner as it uses the existing model to actively pick subsets of instances from unlabeled data which, when labeled, will provide the highest information gain. The strategy should select the instances such that the learning process is sped up when compared to a random selection. We propose two different strategies for selecting instances to be labeled by the crowd:

- **Uncertainty Selection** We choose those instances for which our Automatic Method is most uncertain about the label assignment, depending on how this is computed.
- **Representative Selection** We choose instances such that they cover the entire spectrum of certainty in the assigned labels, depending on how the labels are computed by the Automatic Method.

7.1.7 Improving an Automatic Method

The automatic method produces an *Automatic Label*, $AL(p_{i,j})$ which in the case of deduplication corresponds to 1 for a duplicates pair and -1 for non-duplicates pair. Furthermore the automatic methods can produce an *Automatic Soft Label*, $ASL(p_{i,j})$, which can also serve as an indicator of the confidence the automatic algorithm has in his label assignment. This soft label assignment is used by the *Selection Strategy* to find those instances that will provide the highest information gain to the automatic method when added to the training set.

We propose two types of automatic methods that can learn from the crowd labeled data obtained through the active learning acquisition process.

Duplicates Scorer

The entity matcher, called **DuplicatesScorer**, is introduced in [MBB+10]. For a given pair of entities $p_{i,j}$, the algorithm computes an Automatic Soft Label $ASL(p_{i,j}) \in [0,1]$ as a variant of *ϵ-adjusted geometric mean* of field value similarities based on

$$DSParams = \{(F_k, W_{F_k}) | F_k \in FieldNames\}$$

,where $W_{F_k} \in [0,1]$ denotes the weight of the F_k field in the computation. The final label assignment is computed by comparing the soft label with a chosen *threshold*.

The label provided by the automatic algorithm is

$$AL(p_{i,j}) = \begin{cases} 1, & ASL(p_{i,j}) \geq threshold \text{ ; duplicates} \\ -1, & ASL(p_{i,j}) < threshold \text{ ; not duplicates} \end{cases}$$

When using it as an automatic method in the active learning framework we start with a common sense parameter choice to identify the initial items for which we get labels according to the selection strategy. In the subsequent rounds we learn new parameters for the DuplicatesScorer using knowledge of the parameter choice identified in the previous round. Training this method is done by using an optimization algorithm that finds the parameter choice that maximizes the accuracy on the testing set. The measure of label uncertainty used by the *Selection Strategy* is the distance between the $ASL(p_{i,j})$ and the *threshold*. For this automatic method training is equivalent to optimizing the $DSParams$ and the *threshold* such that the labels provided by it match to those provided by the crowd. We will use the work done by humans to maximize the accuracy of the automatic method. Using the reputation system to put a lower weight on the contribution of the bad workers, we will get as close as possible to a real ground truth, that will be used to compute the accuracy.

For a given pair of entities $p_{i,j}$, the DuplicatesScorer produces a score $ASL(p_{i,j})$. Based on the crowd labels we aim at learning the parameters $DSParams$ and $threshold$ that fit best to how humans solve the deduplication task.

In order to optimize the parameters we use a multi-objective Evolutionary Algorithm (that includes SPEA2 and NSGA2) as implemented in OPT4J [LGRT11] library. We also implemented a Hill Climbing algorithm for optimization, but the results were worse than the ones we obtained using the Evolutionary Algorithm.

We can optimize for several objective functions:

1. Accuracy when comparing the final label assignments for the crowd, CL, with the automatic labels, AL
2. The correlation between the soft aggregated crowd labels, CSL, and the soft automatic labels , ASL
   (a) Mean Absolute Error
   (b) Sum of the log of errors
   (c) Pearson correlation of the two series

If optimizing with respect to Accuracy, we will find both the $DSParams$ and the *threshold* that maximize the objective function

$$Accuracy = \frac{\|\{p_{i,j}|p_{i,j} \in P \text{ and } CL(p_{i,j}) = AL(p_{i,j})\}\|}{\|P\|} \tag{7.5}$$

If we optimize with respect to the correlation between the the crowd's soft label and the Duplicate Scorer soft label, we need to compare $CSL(p_{i,j})$ with $ASL(p_{i,j})$, and find those $DSParams$ that yield the best correlation between the two. After that we seek the *threshold* that gives the highest *Accuracy* as defined in Equation 7.5 .

For the Mean Absolute Error we want to minimize

$$\frac{\sum_{p_{i,j} \in P} \|CSL(p_{i,j}) - ASL(p_{i,j})\|}{\|P\|}$$

For the sum of the log of errors we want to maximize

$$\sum_{p_{i,j} \in P} log(\|CSL(p_{i,j}) - ASL(p_{i,j})\|)$$

For the Pearson Correlation we want to maximize the Pearson Correlation coefficient when comparing the $CSL(p_{i,j})$ and $ASL(p_{i,j})$ for all $p_{i,j} \in P$

With consecutive batches of work done by the Amazon Mechanical Turk workers, the performance of the automatic algorithm will improve, as the choice of $DSParams$ and *threshold* will fit better to the way the crowd does the deduplication.

Classifiers

We can employ a classifier using various features for assigning the labels $AL(p_{i,j})$ to the instances. For a given pair of entities $p_{i,j}$ the feature set used consists of the similarities between the values of the different fields that characterize each entity.

$$Features(p_{i,j}) = \left\{ sim(V^i_{F_k}, V^j_{F_k}) | F_k \in FieldNames \right\}$$

For the similarity employed we propose to use either the Jaccard or the Needleman-Wunch similarity.

When using a classifier in the active learning framework, we start by choosing a random sample of instances for training. In the subsequent consecutive rounds we take into consideration the certainty of labels being assigned to instances in accordance with the selection strategy and the already existing training set, in order to select new instances and improve the performance of the classifier by building a better training set.

7.2 Dataset

Without restricting the generality of our methods, we focus on the domain of digital libraries and present examples of scientific publications. We use pairs of publications labeled either as "duplicate" or "non-duplicate" pairs. The complete list of fields that a publication can have is *FieldNames*={ *Title, Subtitle, By, In, Type, Publisher, Organization, Abstract* }.

The *By* field is composed of person-related data like *Author*, *Editor*, *Contributor*. The *In* is composed of venue related data like *Book*, *Journal*, *Conference*, *Venue* and *Series*. The most discriminative field is of course the *Abstract* because this is distinct for each publication, but not all the publications in our dataset contain it. If all publications would contain the *Abstract* field (which is not usually the case), the task of identifying duplicates would be trivial, being reduced to comparing just this field for the publications in a pair. We have not used the publication year in our experiments since we found this to be mostly unreliable, considering that entries of the same publication could be associated with dates differing by as much as 2 years.

The dataset is composed of publication pairs, that can be labeled as duplicate or non-duplicate pairs. The publications come from 4 different sources: DBLP[2], CiteSeer[3], BibSonomy[4] and TIBKat[5], as presented in the scientific publications search engine *Freesearch*[6] [FGNS11, FGN12, FGNS12]. Having documents from multiple

[2] http://www.informatik.uni-trier.de/~ley/db/
[3] http://citeseerx.ist.psu.edu
[4] http://www.bibsonomy.org
[5] http://www.tib-hannover.de/de/die-tib/opendata/
[6] http://dblp.kbs.uni-hannover.de

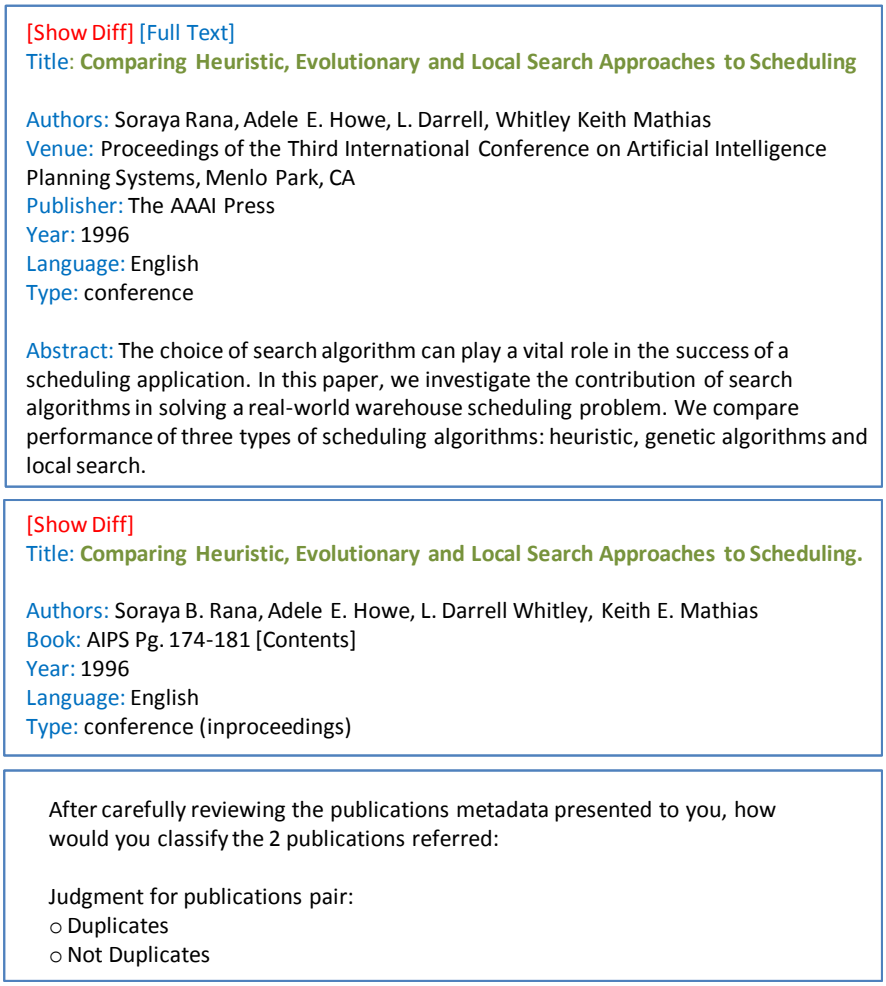


Figure 7.1 Example of a Mechanical Turk Task

sources, with various data quality leads to duplicates, between sources, or even within the same noisy source, to be identified using the proposed methods.

7.2.1 Data Gathering

Focusing on the domain of digital libraries, we create HITs consisting of 5 publication pairs for which the workers have to assign the "duplicate" or "non-duplicate" labels. Detailed instructions are provided, and in addition examples help the workers to understand and solve the task better. Moreover, the worker can see the differences between the two publications, highlighted in different colors. Each HIT pays 5ct. to the worker that solves it. We start with 3 assignments, and if we need to get a more categorical decision from the workers for a pair we extend the HIT that contains it by adding more assignments. We present an example of how a publication pair was

shown to the crowd workers in Figure 7.1.

The *Ground Truth* was manually labeled by 3 experts and is composed of 363 pairs, of which 101 are considered duplicates and 262 non-duplicates. These pairs were selected such that they cover all the workers, in order to also use them to manually asses the worker confidence. The *Crowd Data* was labeled using Amazon Mechanical Turk, and it is used for improving the automatic methods by using the active learning label gathering strategy. It consists of 2070 pairs with at least 3 corresponding labels each, out of which 570 pairs have 7 labels each. According to the majority vote the 2070 pairs are split into 804 duplicate pairs and 1262 non-duplicate pairs.

We have gathered the data in two steps. The first step was driven by doing an initial check [GPF+12] for the feasibility of using our method with the Duplicates Scorer, and in the second step extended the dataset for a more in-depth analysis [GPF+14] of our proposed methods.

Initial Dataset. We use Amazon's Mechanical Turk as a crowdsourcing platform. Each HIT consists of 5 pairs of publications. We start by posting a batch of 60 HITs having a qualification test as prerequisite, and a batch containing 60 HITs and one containing 119 HITs without the qualification test. The pairs sent were selected on the following criteria: $ASL(p_{i,j}) \in \{0.7, 0.8\}$ obtained with the DuplicatsScorer using as parameters *DSParams*={*(Abstract, 0.5), (Title,1.0), (Organization, 0.5), (By, 1.0), (Type, 0.5), (In, 1.0), (Publisher,0.5)*} and $threshold = 0.75$. This serves as our initial parameter choice, that will generate the first candidates.

In this first step of data gathering we retrieved a total number of 1,132 assignments from Amazon Mechanical Turk, corresponding to 239 HITs, for 1,195 pairs.

The HITs were solved by 78 unique Mechanical Turk workers. The average time per HIT was 90 seconds. The average time for solving a HIT for a worker was of 145 seconds. The average number of HITs solved by a single worker was 72. To compare, our experts, assigned labels to 150 pairs in 60 minutes, equivalent to an average of 2 minutes per HIT.

After computing the user confidence against the 363 pairs that compose our ground truth, the average worker confidence was 0.85 with a standard deviation of 0.14. We wanted to see if the confidence in the worker correlates with the average time he invests in solving the HIT. The Pearson Correlation Coefficient between the average time per HIT and the worker confidence is 0.177, which indicates a very low correlation. The worker confidence also does not correlate with the number of HITs a worker solved, the Pearson Correlation being -0.19. Finally we investigate the correlation between the average time for solving a HIT and the number of HITs solved, and the value of -0.25, shows that there is a low negative correlation. That indicates that there is no correlation between the average time a worker dedicates to solving a HIT, the number of HITs he solves, or the confidence we have in him.

Table 7.1 Accuracy, number of pairs used for optimization, threshold and weights for DuplicatesScorer learned in consecutive Active Learning rounds in the data gathering process.

Rnd.	Accuracy	Pairs	Thresh.	Abstract	Title	Subtitle	In	By	Type	Org.	Publ.
0	0.77027	-	0.75	0.5	1	0.5	1	1	0.5	0.5	0.5
1	0.78815	570	0.72	0.17	0.98	0.2	0.05	0.3	0.19	0.01	0.26
2	0.78981	500	0.73	0.55	0.97	0.1	0.27	0.37	0.21	0.0	0.01
3	0.79504	500	0.75	0.7	0.98	0.1	0.47	0.54	0.13	0.0	0.04

Extended Dataset. In order to perform experiments on the resource allocation and selection strategies, we extended the previously presented dataset. The data was gathered in 3 rounds, guided by a deployment of our algorithm with the Duplicates Scorer as an automatic method, and the uncertainty selection strategy. To test the hypothesis that with each round of active learning, our algorithm will exhibit an improvement and need less and less input from the crowd, our algorithm is run for 3 rounds. The accuracy of each run is presented in Table 7.1, along with the number of pairs included in the sample used for training. The different weights for the fields reported by each run are also presented. For Round 0, the starting point of the active learning process, we used a common-sense parameter choice, in order to be able to compute initial scores and select a sample of pairs with an uncertain assignment. This coincides with the choice made for the initial dataset. Of the 1,195 pairs from the initial dataset, after excluding the 363 pairs that compose the ground truth and the pairs for which only the year differed, we are left with 570 pairs that are used for optimization in Round 1. In the following rounds we always start optimizing from the parameters we learned in the previous round. The pairs for the forthcoming round are selected so that their score computed using the current weights is around the threshold of the current round.

The weights change between rounds, but remain close to the ones computed in the previous rounds. The learned threshold remains in the proximity of the common-sense threshold that we started with. The accuracy slightly increases, validating our hypothesis, that the parameters learned in consecutive rounds provide better performance.

For the experiments on the optimal number of tasks to be solved in a round, we used the same parameters as in round 1 to get 500 more pairs, in order to ensure pairs are selected in the same way. For the experiments on the optimal number of assignments, we extend the 570 pairs having 3 labels from the cleaned initial dataset, by gathering 4 more labels for each of them. Therefore, we extended the initial dataset by requesting 4 more labels to the original 570 pairs, then, based on the same parameter choice, we collected 500 more pairs with 3 labels and finally in 2 consecutive active learning rounds 1000 more pairs with just 3 labels.

To investigate any link between the importance of the attributes and their distribution in the dataset, in Figure 7.2 and Figure 7.3 we present a histogram of the number of pairs and field names for which a pair has values for either both, none, or

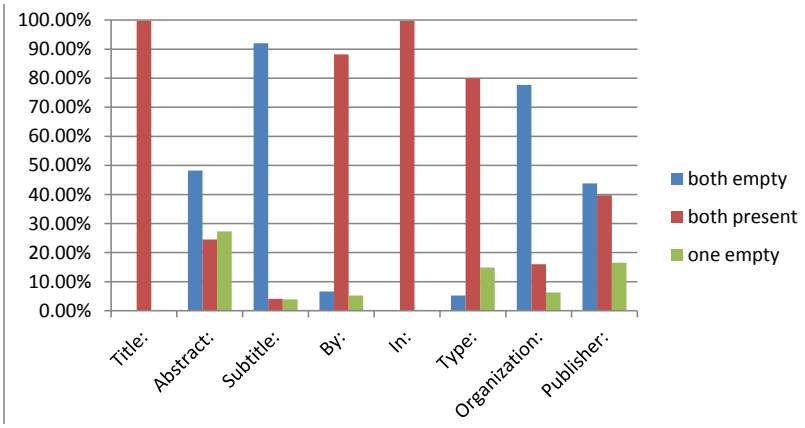


Figure 7.2 Analysis of attribute distribution in the Ground Truth data

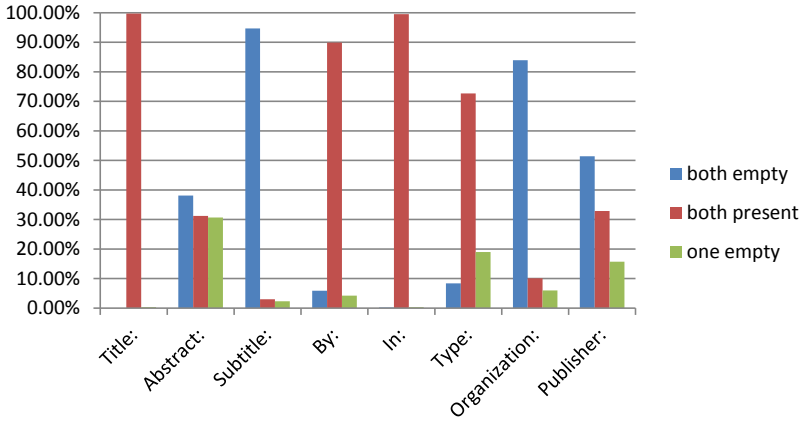


Figure 7.3 Analysis of attribute distribution in the Crowd data

only one field name, in the ground truth as well as in the crowdsourced data.

The distribution for the Ground Truth is similar to that of the Crowd Data. Furthermore, we can see that all of the pairs have the *Title* field present. The other fields that appear in both publications of most pairs are *In* , *By* and *Type*. The *Subtitle* and *Organization* fields are in most cases empty for both publications in the pair. *Publisher* and *Abstract* are well distributed among the classes. Only a small proportion of pairs have the most discriminative field, the *Abstract*.

7.2.2 Agreement between Labelers

In this section we present statistics regarding the agreement between annotators in terms of Fleiss' Kappa and Krippendorf's Alpha. In Section 7.5.1 we will discuss further about the correlation between the annotator agreement and the performance of our methods. In Table 7.2 we can see the number of instances that have as many assigned crowd labels, as well as the two indicators for measuring the agreement

Table 7.2 According to the number of labels available(Labels), the number of instances in the dataset having at least that number of labels (Instances), and two agreement indicators between the labelers: Fleiss' κ (κ) and Krippendorf's α (α).

Labels	Instances	κ	α
Experts Agreement on GT			
3	362	0.827	0.827
Crowd Agreement on GT			
3	358	0.526	0.526
4	358	0.526	0.526
5	358	0.503	0.511
6	337	0.478	0.499
7	285	0.47	0.492
Crowd Agreement on all data			
3	2064	0.282	0.282
4	570	0.506	0.303
5	570	0.499	0.319
6	570	0.495	0.331
7	570	0.477	0.338

between annotators. We can clearly notice that the agreement between the expert assessors for the ground truth is the highest. On evaluating the agreement between crowd labelers on the ground truth, we notice that contrary to our expectations, introducing more labelers actually decreases their agreement. We do not have the same number of instances for each choice of number of labelers, but 3 assessors are clearly in more agreement than 5. In contrast, on evaluating the agreement of the crowd on all the data, we see that introducing more than 3 labelers actually increases the agreement, 5 workers resulting in more mutual agreement than just 3. Nevertheless, in this setting introducing more than 5 workers also has a detrimental effect on the agreement. The agreement of the crowd on the ground truth data is comparable to the agreement on the whole data, except in the case where we employ 3 workers.

When comparing the labels where the crowd labelers did not agree with one another and the ground truth we notice several patterns. Most significantly the crowd labelers assign the duplicate label to non-duplicate pairs, by investigating only superficially the title, and ignoring the other fields in order to produce their labels.

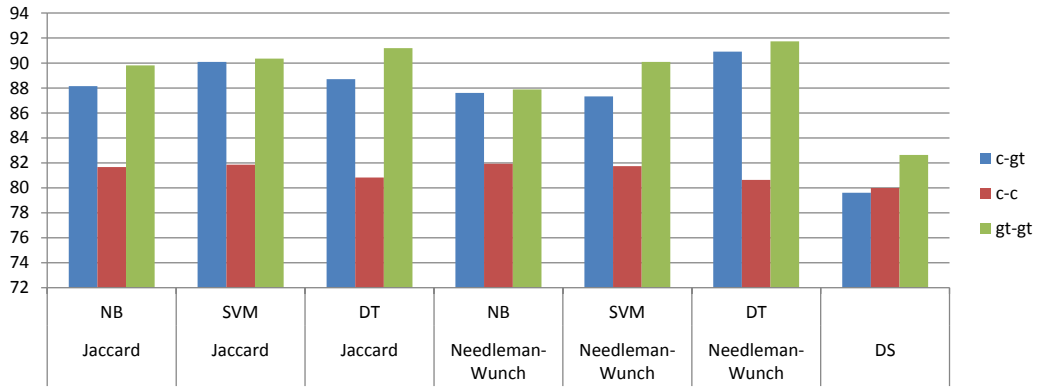


Figure 7.4 Accuracy of the different methods on various settings.

7.3 Experiments

7.3.1 Accuracy for Various Automatic Methods

As a first experiment we assess the performance of the different Automatic Methods: classifiers and Duplicates Scorer. We used Majority Voting as a label aggregation strategy.

The chosen classifiers are Naïve Bayes(NB), SVM(SVM) and Tree(DT). We used their respective Weka [HFH+09] implementations: Naïve Bayes, SMO and C4.5 (J48) with the default parameters. A pair of $p_{i,j}$ is characterized by 8 features $sim(V^i_{F_k}, V^j_{F_k})$ where $F_k \in \{Title,\ Subtitle,\ By,\ In,\ Type,\ Publisher,\ Organization,\ Abstract\}$ and sim can be either the Jaccard similarity based on tokens or the Needleman-Wunch similarity at character level.

The parameters of the Duplicates Scorer (DS) are the weights of the fields describing the publications *Title, Subtitle, By, In, Type, Publisher, Organization, Abstract*. They are used to compute $ASL(p_{i,j})$ and the *threshold* for deciding if the pair is a duplicate or non-duplicate pair. As a learning process for the Duplicates Scorer we used the accuracy maximization optimization strategy.

In Figure 7.4 we report the accuracy for different settings: training on the crowd data and testing on the ground truth(c-gt), 10-fold cross-validation when training and testing on the ground truth (gt-gt) and 10-fold cross-validation training and testing on the crowd data (c-c).

We notice that in all the settings the Duplicates Scorer is surpassed by the classifiers, and among the classifiers, the decision tree seems to provide the best results. We also observe that the different settings for the training and testing data have different accuracies. We do not observe any remarkable differences between using the Needleman-Wunch or Jaccard as a similarity measure. The highest accuracy is obtained when training and testing on the ground truth data. This might be due to the high quality data, which is also reflected in the agreement statistics presented in

Table 7.3 Attribute selection.

Field	Leave-1-out	Chi-squared	Info gain
Title	0.737	671.510	0.352
Subtitle	0.790	0	0
Abstract	0.796	156.448	0.076
By	0.782	163.730	0.081
In	0.788	89.198	0.042
Type	0.781	0	0
Organization	0.793	2.665	0.001
Publisher	0.792	29.074	0.013

Section 7.1.3. Training on the crowd data and evaluating on the ground truth also results in a good performance. The lowest performance is reported when training and testing on the crowd data. The lower performance obtained when using crowd data for testing might be due to its inherent noisy nature or lower quality.

7.3.1.1 Attribute Selection

In order to determine which fields are more important for different automatic methods we used 3 attribute selection methods. The results are presented in Table 7.3. For the Duplicates Scorer we ran 'Leave one field out' experiments and reported the accuracy achieved; the lower the accuracy is, the more important the field is. We can see that in this case the most important fields are: *Title, Type* and *By*. For assessing the importance of the fields when using classifiers we conducted a chi-squared test and evaluated the information gain; the higher the score, the more important the field is. For both tests *Title, By* and *Abstract* were identified as the most important fields.

For all the different automatic methods the *Title* is the most discriminative field, but the next fields of secondary importance depend on the method. The Duplicate Scorer considers *Type* more important then *By*, while the classifiers consider the *Type* information as being unimportant. The classifiers also consider the *Abstract* (which is actually the most discriminative field) as important, although it is not present in many of the publication pairs.

7.3.1.2 Peculiarities of Automatic Methods

The maximum optimized accuracy for the Duplicates Scorer that we can achieve on the ground truth is 0.83. This corresponds to the following learned weights: *Abstract=0.1, Title=0.9, Organization=0.1, Subtitle=0.1, By=0.3, Type=0.8, In=0.3, Publisher=0.0* and *Threshold=0.76*. This is consistent with the results obtained using the 'leave one field out' experiments in Section 7.3.1.1. The fields with the highest weights are the *Title*, *Type* and *By*.

The rules learned by the best performing classifier confirm the importance of the attributes that we examined in Section 7.3.1.1. When using the Jaccard similarity for both *c-gt* and *c-c* the learned tree first compares the *Title* similarity, then the *Type*, *Publisher* and finally the *By* similarities. For the *gt-gt* setting the classifier compares just first the and then the *Publisher*. When using the Needleman-Wunch similarity for both *c-gt* and *c-c*, the learned tree first compares the *Title* similarity, and then either the *Abstract* or *In* similarities. In the *gt-gt* setting the only rule is to compare the *Title* similarity.

7.3.2 Resource Allocation

The resources to be allocated for a round of active learning from the crowd, are the number of pairs from which we build the HITs to be posted on MTurk (indicated by the sample size s), and the number of assignments per HIT (corresponding to the number of workers that provided labels for an instance). In this section we explore the effects of different levels of resource allocation on the performance of the automatic method, the Duplicates Scorer after learning the optimal parameters, and the classifiers after retraining respectively.

7.3.2.1 Number of Assignments per Task

We use the majority voting as a label aggregation strategy. This is similar to considering that all the workers have the same confidence. To determine the optimal number of labels we need for a pair to be accurately adjudged on MTurk, we experiment on a batch of 570 pairs having 7 labels and we compute the accuracy obtained by taking into consideration only the first workers that assigned labels on each pair. We plot the accuracy obtained by using different Automatic Methods for different number of assignments in Figure 7.5.

In the case of using the Duplicates Scorer, the performance we gain by introducing more than 3 workers is of only 1% for 7 workers and negligible when using 5 workers. Although the difference in accuracy between the setting where we use 1 worker and the setting with 3 workers is of 2%, we cannot use just 1 worker since on doing so we risk relying on only one opinion, which might be biased.

In the case of using classifiers we notice a drop in performance when using 5 workers when compared to using just 3 workers, that is recovered when using 7 workers for Naïve Bayes and Decision Trees. For SVM the best performance is obtained when using just 1 worker, and using 3 causes a drop that is slowly recovered by adding additional workers.

For almost all the considered automatic methods considered, the additional cost of adding 2 or 4 more labels when we already have 3, is not justified by the performance increase. Therefore, 3 is optimal number of workers for the task of gathering labels for deduplication scientific publications.

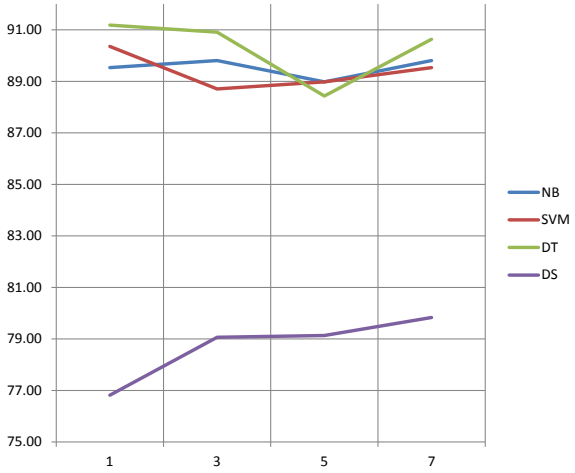


Figure 7.5 Number of assignments per task vs. accuracy of the automatic methods.

7.3.2.2 Number of Instances per Active Learning Round

To determine the optimal size of the sample that has to be labeled by the crowd in an active learning round, we experiment with 1070 pairs of training data obtained by using just 3 workers. We regard these workers to be equally competent, and thereby the crowd label is based on majority voting. According to this label aggregation strategy 408 pair are duplicates and 662 are non-duplicates. We select samples of pairs of different sizes containing a number of duplicate or non-duplicate pairs proportional to the number of overall available pairs. We present the average accuracy obtained from 20 rounds of randomly sampling the pairs and retraining. We plot the accuracy obtained on optimizing for different number of pairs in Figure 7.6. For all the automatic methods we notice a considerable increase in performance for up to 500 examples per round; going over this number of examples does not result in an improvement that would justify the extra cost.

7.3.3 Eliminating Unreliable Workers

We investigate the effect of eliminating the workers that do not provide reliable assignments, on the accuracy of the Automatic Method. Instead of considering all workers as being equally good at deduplication, we compute the worker confidences using Algorithm 3. Thus, the labels of different workers will be weighted by the confidence we have in him. We experiment with eliminating the workers whose computed confidence is under the reliability threshold, $WQThreshold$. The dataset used for this experiment is composed of 570 pairs, each having 7 labels from workers with different confidences. We optimize the parameters of the DuplicatesScorer using the pairs and labels that remain after eliminating the unreliable workers. We compare the effect of

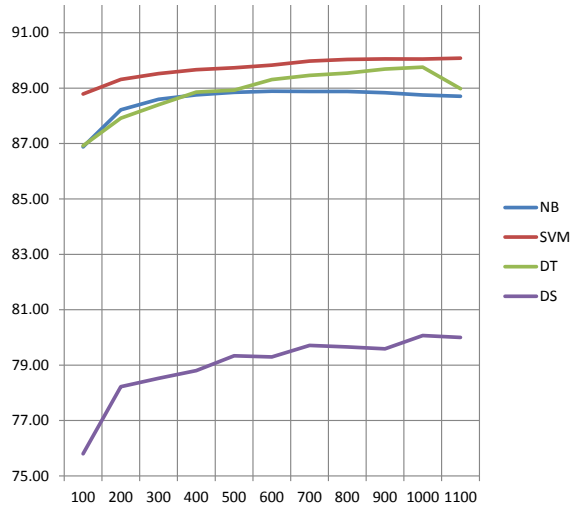


Figure 7.6 Number of tasks for each round vs. accuracy of the automatic methods.

using different thresholds on the accuracy, obtained by comparing the labels provided by the crowd to labels assigned by experts contained in our ground truth. For each threshold we compute the accuracy, show how many pairs are consequently left for optimization, and how many workers exhibit a confidence greater than the threshold, as presented in Table 7.5. We present the distribution of the number of labels for the available pairs in Table 7.4.

On examining Table 7.5 we notice that the maximum accuracy for the Duplicates Scorer is obtained when eliminating all the workers having a confidence under 0.9. This retains a reasonable number of pairs, based on which the optimization of the DuplicatesScorer can be carried out, while filtering out half of the workers. Although the number of pairs is consistent, by looking at Table 7.4 we see that most of these pairs have only 3 labels assigned to them. Comparing to the setting where we use just 3 labels, and do not eliminate any of the workers the gain in accuracy is of only 1%, but the costs are much higher, for obtaining the extra labels that will be discarded. Thus, in this setting using just 3 labels per pair without eliminating the poorly performing workers, can be recommended. The workers identified as unreliable should only be prevented from participating in the next rounds.

For the classifiers, the best thresholds for eliminating workers $WQThreshold$ is quite low, 0.7 for Naïve Bayes and the Support Vector Machine classifiers, while for the Decision Trees a threshold of 0.65 or even 0.6 seems to provide a good performance. The low threshold allows more workers to participate in the decision process for the aggregated crowd label. This is not considered by the classifiers as contributing noise, but as a means to improve their performance. The classifiers are more robust and relatively independent of the worker quality threshold, and at the same time they show superior performance when compared to the Duplicates Scorer.

Table 7.4 The distribution of the number available labels according to each selected worker quality threshold.

QT	Number of available labels						
	1	2	3	4	5	6	7
0.6	0	0	4	47	64	113	342
0.65	0	0	4	47	64	119	336
0.7	0	0	4	67	208	224	67
0.75	0	65	220	181	95	8	1
0.8	13	90	233	151	78	4	1
0.85	145	185	96	77	43	1	0
0.9	172	236	100	34	1	0	0
0.95	158	15	0	0	0	0	0
1	113	9	0	0	0	0	0

Table 7.5 Statistics on the pairs samples after eliminating unreliable workers. The Accuracies obtained by different methods: Naïve Bayes (NB), Support Vector Machines(SVM), Decision Tree(DT) classifiers and the Duplicates Scorer(DS), along with the Number of available pairs for training (P) and the numebr of workers(Wrk) contributing labels to those pairs, according to the selected worker quality threshold(QT).

QT	NB	SVM	DT	DS	P	Wrk
60	89.81	89.81	90.08	79.66	570	88
65	90.08	89.81	91.18	79.64	570	84
70	90.08	90.08	89.53	79.35	570	81
75	89.26	89.81	90.63	80.47	570	77
80	88.43	89.81	90.36	80.26	570	73
85	89.26	89.81	88.98	80.48	547	52
90	89.26	89.81	88.98	80.99	543	42
95	84.30	87.33	87.60	79.30	173	28
100	84.30	85.40	90.63	80.00	122	24

7.3.4 Candidate Selection Strategies

In this section we experiment with the entire dataset that was labeled by the crowd by simulating the process of acquiring the labels from the crowd. The label aggregation method employed is also Majority Voting. In each step we procure labels from the crowd for corresponding $s \in \{10, 20, 50\}$ instances and we apply our method. We have chosen to use three types of automatic methods that we want to improve: Duplicates Scorer and Naïve Bayes using the Jaccard similarity or the Needleman-Wunch similarity. We plot the Accuracy obtained at each step, using different sampling strategies for the instances that are to be labeled. The different candidate selection strategies we use in our experiments are *Uncertain*, *Random*, and *Representative*.

The *Representative* strategy divides the pool of unlabeled instances into bins according to the uncertainty of their labels. Bin B_u contains instances with an uncertainty in the $[0.1 \cdot u, 0.1 \cdot (u + 1)]$ interval, with $u \in \{0, 1, .., 9\}$, covering in this way the whole uncertainty interval $[0, 1]$. Thereafter, in each round of candidate selection it selects a number of random instances from each bin that is proportional to the binsize. In this way we do not choose the most uncertain instances, but instances with representative uncertainty.

If we use the Naïve Bayes classifier as the automatic machine learning algorithm: *Uncertain* refers to getting the s pairs that have a probability to belong to one class closest to the uncertainty threshold (0.5). *Random* chooses s pairs randomly from the entire dataset. For the *Representative* strategy the measure of uncertainty is computed as the difference between the probability of belonging to the negative class and the probability of belonging to the positive class. We experiment with both similarity metrics for computing the features: Jaccard and Needleman-Wunch.

In the case of the Duplicates Scorer as the automatic algorithm that we propose to use in our method, *Uncertain* refers to getting the s pairs that have a ASL that is closest to the learned threshold for that step. *Random* chooses s pairs randomly from the entire dataset, independent of the ASL. For the *Representative* strategy, the ASL is the used measure of uncertainty.

We report the accuracy of the automatic method after retraining for each round of active learning in terms of learning curves in Figure 7.7. For the classifier *Random* performs better than the *Uncertain* strategy, pointing to the fact that a strategy that takes into account the representativeness of the instances would yield better results. Using the *Uncertain* strategy, the crucial example that improves the performance significantly is discovered later than in the *Random* case. Of course we notice an improvement of the performance as more instances become available with each round of label gathering. As expected, this example is found early by the *Representative* selection strategy. The *Representative* strategy performs better than the *Uncertaint* strategy, but it is not clearly better than the *Random* sampling.

In the case of using the Duplicates Scorer as the automatic algorithm that learns from the crowd, we see that using 10 or 20 pairs is not efficient for both *Random*

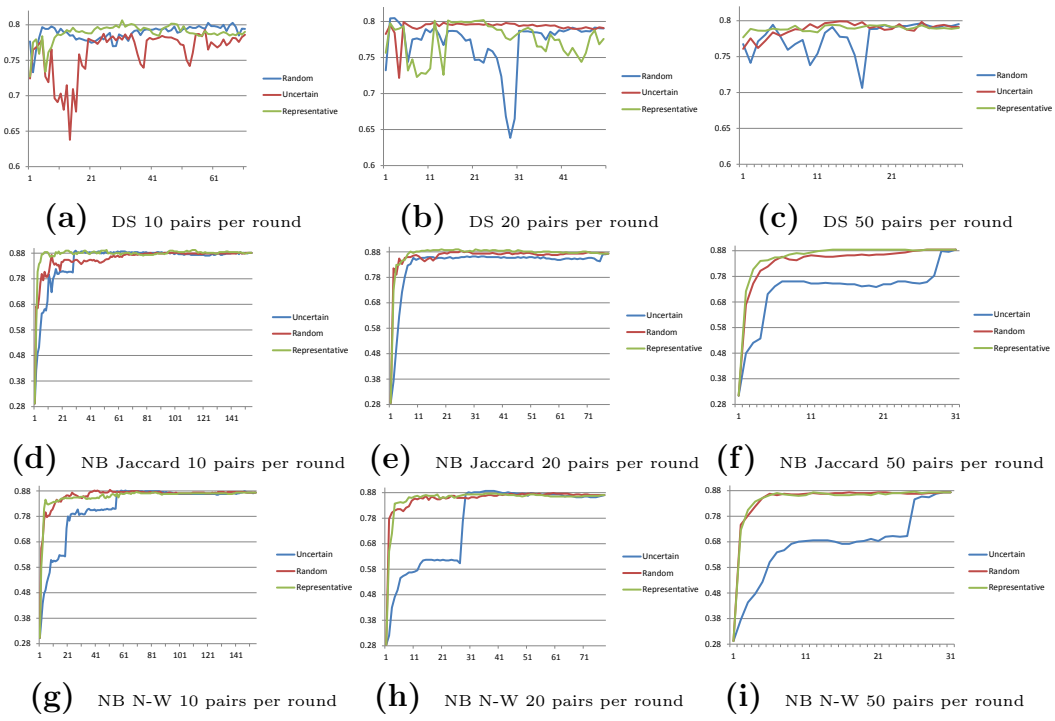


Figure 7.7 Learning curves for active learning on crowd data collected using different automatic methods, candidate instance selection strategies and number of instances for each active learning round

and *Uncertain*. We do not see an increase in performance with each round as we would expect. This is different for 50 examples per round, we see the increase in performance, and also the *Uncertain* strategy beats the *Random* sampling. We can conclude that 10 and 20 are a small quantity of new labels for our method to exhibit improvement. This effect can only be seen from upwards of 50 instances. In the resource allocation experiments presented in Section 7.3.2, we observed that a good number of instances per round is around 500.

7.3.5 Employing the Duplicates Scorer

In the case of majority voting the all workers have the same confidence, equal to one. We can use other strategies by employing the worker confidences directly to compute the aggregated label, or by boosting them so that workers with a higher confidence have a bigger importance in the aggregation process for a pair.

We use the following notations: CSL_{MV} represents the soft label in the case of majority voting, when $w_k = 1, \forall k$, specifically $CSL_{i,j} = (1 + \sum_{k \in W_{i,j}} W_k(p_{i,j}))/2$; CSL_{iter} represents the soft label computed by using the worker confidences outputted by the Algorithm 3; CSL_{iter}^{boost} appears when we use the boosted weight in computing the soft labels using the same Algorithm 3. We also have the possibility to manually

Table 7.6 Duplicate detection performance with respect to the optimization method, and the strategy for reaching the .

Optimization	Strategies																	
	3 workers			5 workers														
	CSL_{MV}			CSL_{MV}			CL_{heur}			CSL_{iter}			CSL_{iter}^{boost}			CSL_{manual}		
	A	P	R	A	P	R	A	P	R	A	P	R	A	P	R	A	P	R
Accuracy	0.79	0.66	0.56	0.80	0.68	0.55	0.80	0.67	0.56	0.80	0.67	0.56	0.80	0.65	0.57	0.80	0.67	0.58
MAE	0.76	0.56	0.75	0.79	0.64	0.62	0.79	0.63	0.64	0.79	0.64	0.62	0.79	0.64	0.62	0.79	0.64	0.62
$Sum - log - err$	0.72	0.50	0.60	0.78	0.63	0.54	0.77	0.60	0.55	0.78	0.64	0.53	0.80	0.70	0.53	0.79	0.65	0.55
Pearson	0.73	0.52	0.82	0.79	0.62	0.70	0.81	0.69	0.61	0.79	0.62	0.70	0.79	0.62	0.70	0.81	0.67	0.61

asses the worker confidences, on a sample of pairs for which we know the ground truth. We will name this CSL_{manual}. Each of these aggregated crowd soft labels correspond to a hard label : CL_{MV}, CL_{iter}, CS_{iter}^{boost} and CSL_{boost}.

Apart from computing the crowd label as formalized above, we can use heuristics to compute the final crowd label. For example, the pair will be regarded as a duplicate if all the initial three workers agreed on that. Otherwise, we request two more labels and take a decision if at least four out of five workers agree. Otherwise the pairs are not regarded as duplicates. We will name this heuristic hard label CL_{heur}

On the initial dataset we experiment on using various ways of obtaining the final crowd labels, combined with different optimization strategies, in order to determine the scenario that offers the best results. We present these results in terms of accuracy (A), precision (P) and recall (R) in Table 7.6. We compare the case where 3 workers were used for each pair, or 5 workers. Differences in accuracy varying more than 0.01 are statistically significant by means of a one-tail paired t-test with $p < 0.05$.

Comparing the different crowd label aggregation strategies we find that CSL_{iter}^{boost} works best. Highest results are obtained in the case of optimization for methods penalizing high differences between crowd labels and algorithm output, like *Pearson* or $Sum - log - err$. $CL_{i,j}^{heur}$ uses the best data in terms of agreement between workers and achieves very good results. Nevertheless the results are very close to the simple CSL_{MV} strategy using 3 instead of 5 workers when optimizing for *Accuracy*. Similarly, optimizing for *Accuracy* yields very good results consistent across the different aggregation strategies.

Strategies like CSL_{MV}, CSL_{iter}, and CSL_{iter}^{boost} approximate the worker's real confidence. By computing the confidence based on our own assessed ground truth, i.e. CSL_{manual}, we find that although the exact numbers are not completely identical, in our optimization setting this makes little to no difference for the end results.

Compared with the original preliminary parameters choice, the CSL_{heur} strategy, optimizing for Accuracy are *DSParams* ={*(Abstract,0.17), (Title=0.98), (Organization,0.01), (By,0.3), (Type,0.19), (In,0.05), Publisher=0.26*} and *threshold* = 0.72. It is surprisingly to see that the *In* and *By* fields have a very low weight, although one would expect the opposite.

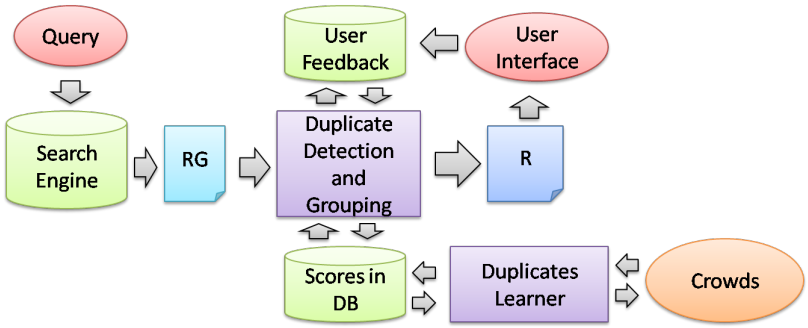


Figure 7.8 Query solving.

7.4 Application to Publication Search

We integrated the proposed approach with the DuplicatesScorer in the *Freesearch*[7] online search system for publications, that aggregates data from different sources: DBLP[8], CiteSeer[9], BibSonomy[10] and TIBKat[11].

In order to offer the user a cleaner view of the results to a search query, the common practice is to group the duplicates together and show just one version. The other versions are available via a link to "all versions". The task of detecting the duplicates and grouping them together is based on signatures, and the version displayed is the one with the highest rank with regard to the used query. Document signatures are Bibliographic Hash Keys[12]: hashes of aggregated normalized basic metadata like author, title, year, venue.

In Figure 7.8 we describe the work flow for resolving a query in our system. The query is forwarded to the Search Engine component and the results pass through the Duplicates Detection and Grouping component to produces a list of results where duplicates appear in the same group. This component takes into consideration user feedback and the DuplicatesScorer scores obtained by using the parameters learned through crowdsourcing.

The way our deduplication methods are integrated in the system are presented in Figure 7.9. A periodical job selects the duplicates that are uncertain and prepares a task for crowdsourcing. After the task is solved, the user feedback and the labels assigned by the crowd are used to learn better parameters for the DuplicatesScorer. The learned parameters are used to recompute all the scores in the database.

In the user interaction with the system, the duplicates detection component comes in more places: for all the queries, for different versions queries, and for similar

[7] http://dblp.kbs.uni-hannover.de
[8] http://www.informatik.uni-trier.de/~ley/db/
[9] http://citeseerx.ist.psu.edu
[10] http://www.bibsonomy.org
[11] http://www.tib-hannover.de/de/die-tib/opendata/
[12] http://www.gbv.de/wikis/cls/Bibliographic_Hash_Key

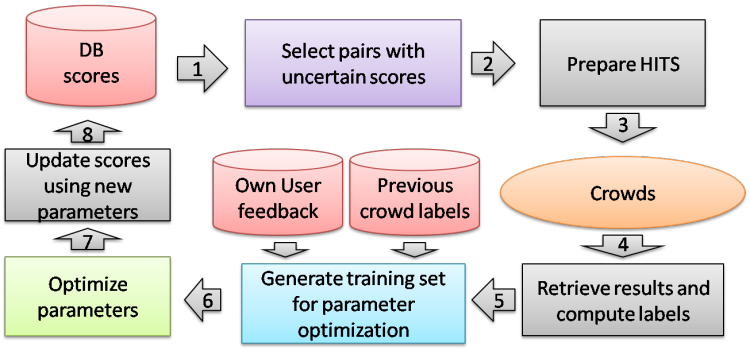


Figure 7.9 Learning how to deduplicate.

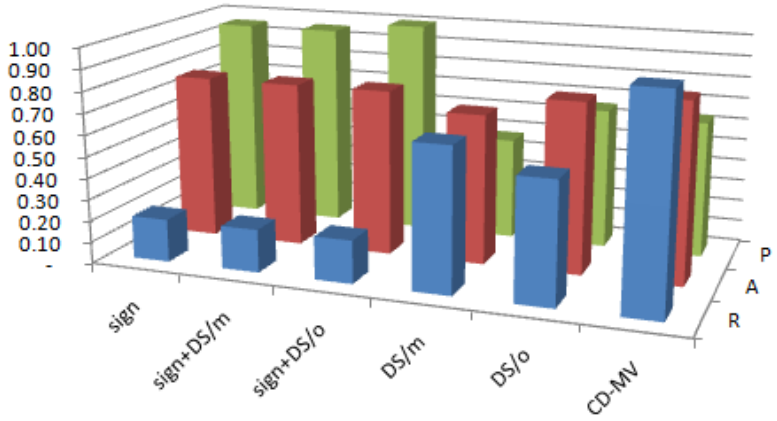
documents queries. In the latter cases, the user can also correct the assignments, by directly giving feedback. The users are motivated to use the feedback mechanism when they inspect their own publications and notice that the duplicates assignments are not correct. This feedback is used to improve the particular users' own display of the search results, but can also help other users. In addition to the feedback received through our own interface, we also take advantage of other ways of involving humans in the process. We resort to crowdsourcing in order to clarify the situations where an assignment as duplicates is not reliable. The crowdsourced work is used directly, to disambiguate results on which we are not certain, and it is also used to improve our automatic component, by learning of better parameters.

We present the following use-cases for using deduplication in query solving:

User Query The user issues a query q. A list of groups $RG = G_i$, containing publications grouped based on signatures is received from the index. If G_i contains just one publication p_i then we add this to the result list to be served to the interface R. For the groups G_i containing more than one publication we split it into S_i and D_i, documents that are duplicates or just similar with the first publication of the group p_i. We make $G_i = D_i$. The first publication in the group p_i is added to the result list R, and a link with a query for the different versions contained in G_i is presented in the interface. Finally all the publications in S_i will be added to the results list R.

Other Versions Query The user queries for other versions of document with duplicates. A query for the publications that have the same signature as the result p_i is issued. We split the R set in a set D of publications that are duplicates of the initiator p_i, and a set S of publications that are only similar and we display the two sets, with the feedback mechanism in place.

Similar Documents Query The user issues a query for publications that are similar to the selected result p_i. The results $R = \{p_r\}$ are computed and retrieved by the search engine managing the index. R is split into D and S, and displayed correspondingly, with the option to give feedback.



	sign	sign+DS/m	sign+DS/o	DS/m	DS/o	CD-MV
R	0.20	0.20	0.20	0.67	0.56	0.97
A	0.77	0.77	0.77	0.70	0.79	0.83
P	0.95	0.95	1.00	0.48	0.66	0.63

Figure 7.10 Performance of various duplicate detection strategies.

To split a result set R into a set D of publications that are duplicates of the query initiator p_i, and a set S of publications that are only similar, but not duplicates we proceed in the following manner. We examine the status of each pair formed by the query initiator document p_i and the other results p_r. For all $p_{i,r}$ if the scores $ADS_{i,r}$ are not already computed, we compute and insert them into a database, otherwise we retrieve them from the database. If the current user gave feedback with respect to $p_{i,r}$ then that is taken as the decision to be used in the interface. If the other users gave feedback on the pair, then the majority vote counts. In the case that there is no user feedback, the Duplicate Score of $p_{i,r}$, will be taken into consideration, and the signatures will be updated as to coincide. For each result p_r, the user of the system has the possibility to give a feedback on $p_{i,r}$, confirming or infirming the output of our algorithm.

Instead of pre-computing the scores for all possible pairs of duplicates we do it just for the ones that come close to the user interaction with our system. As you might have noticed in the use cases presented in the previous section, the duplicates scores are computed just when they are needed. To compute the scores for all the publications would be computationaly expensive, and it will also be infeasible, as the parameters of the scorer can change after receiving feedback from our own users or from the crowd.

7.4.1 Duplicates Scorer in the Integrated System

We implemented different duplicate detection strategies in the *Freesearch* online scientific publication search system. Figure 7.10 shows how the different methods perform in terms of accuracy (A), precision (P), and recall (R). *sign* detects duplicates based only on the publication signature ; DS/m uses the default manual weights and threshold for the DuplicateScorer while DS/o uses the optimized, learned weights using the simplest, most cost-effective strategy, CSL_{MV} with 3 workers optimized for *Accuracy*. $sign + DS/m$ and $sign + DS/o$ are very computationally efficient combined methods: first they group duplicate candidates by signature (for efficiency) and then base the duplicate detection on DS/m or DS/o respectively (for best accuracy). $CL - MV$ is simply the crowd decision out of 3 workers using majority voting – this is the performance of humans given this task.

We have shown how automatically learning features for a general purpose duplicate detection algorithm using a very simple and cost-effective approach increases accuracy from 0.70 (DS/m) to 0.79 (DS/o). Taking a look at $CL - MV$ we see that even humans perform only 4% better, with an accuracy of 0.83.

While integrated in the overall system, $sign + DS/o$ shows a perfect precision at the cost of recall. Still the improvement between $sign + DS/m$ and $sign + DS/o$ is of 5% in precision with no loss in recall or accuracy. Thus, the system presents the user with clean, non-duplicate results, while not showing all possible duplicates of a result – which for most users will not make any difference.

7.5 Lessons Learned

7.5.1 Agreement Influence

As noticed in Section 7.3.1 the performance of the automatic method is dependent on the training and testing data. We assume this has something to do with the difference in quality that can also be noticed in the agreement between the labelers as reported in Section 7.2.2. In the *c-c* setting, we observe that when the training and testing on the aggregated crowd labels, the performance is much lower than in the other cases. The agreement on the crowd data when using 3 labels in the experiment in terms of Fleiss' κ and in terms of Krippendorfs's α are portrayed in Table 7.2. This is much lower than the agreement of the experts on the ground truth data that is used, respectively. By comparing the agreement of the experts with the agreement of the crowd labelers, we observe that the crowd labelers agree less, maybe because of their lack of domain knowledge.

One other concern that is raised regarding the agreement between labelers, is related to the instances that are retained in the loop until the aggregated label surpasses a desired level of confidence. There might be some instances for which assigning a la-

bel is a hard task, on which the different assessors cannot reach an agreement. These instances would remain in the loop for a long time and waste resources in vain. Our method would benefit from a way to identify hard to label instances. These kind of instances could be directed towards expert labelers instead of the crowd.

7.5.2 Comparison of Automatic Methods

We can notice a clear inferiority of the Duplicates Scorer when compared to employing classifiers. The Duplicate Scorer was developed for a setting where the attributes are unknown a priori. It was intended to be deployed in case neither the type of entity nor the attribute labels to be used in order to describe the entities are known. In such scenarios, machine learning methods cannot be used, because the features are unknown. The threshold can however be learned from the crowd, as shown in [GPF+12]. For the scenario in the current chapter, where the attributes (features) are known and fixed, the best approach seems to be employing classifiers. The advantage of the Duplicates Scorer is that it is applicable to unknown entities, at the cost of a decrease in accuracy.

Selecting an appropriate method that learns from the crowd is very important, because it influences the candidates selection strategy. We cannot use a selection strategy based on one method to train another method. As each new label has an associated cost, the right method and the right selection strategy both play a key role in achieving the desired performance within the required budget and time.

7.5.3 Crowd vs. Experts

The results of crowdsourcing are strongly influenced by the difficulty and complexity of the task. Although the task we want to solve is easy at first glance, being reduced to comparing different fields, it is not a trivial task and even the experts had disagreements. We have discovered that it is not as trivial for crowd workers as it is for people with domain knowledge. This is evident when we look into the agreement between the crowd workers and compare it to the agreement between experts.

Examining Table 7.2 we notice that the agreement between the crowd workers is much lower than the agreement between experts. We have also investigated the most common reasons why the crowd workers assign the wrong labels, or where they disagree the most. We conclude that crowd workers, not possessing enough domain knowledge, oversimplify the task, and only compare the titles of the publications. They completely ignore the other fields, and they superficially compare the titles. Domain knowledge such as recognizing the abbreviations of conference titles, or the patience the experts have proven to exhibit are crucial in assigning the correct labels.

7.5.4 Data Distribution

If we examine the label distribution in the ground truth we can notice a high unbalance towards pairs that are not duplicates. Thus just by assigning the non-duplicate label, we could achieve an accuracy of about 70%. If we consider majority voting as the label aggregation strategy, the crowd data is also unbalanced consisting of 61% non-duplicate pairs and 39% duplicate pairs. In this particular scenario, detecting duplicates in a publication database, this is not unusual, reflecting the real situation where the number of duplicates depends on the number of sources and their quality.

The field distribution presented in Section 7.2 could also offer some insight into the importance of the attributes. The fields that appear in both publications of most pairs are: *Title*, *By*, *In* and *Type*. The other fields are either both empty or one of them is empty. As expected, those fields that are mostly empty for both publications of a pair are identified as the least important in the attribute selection experiments and also have the lowest weight for the learned Duplicate Scorer. These fields are not mentioned at all in the rules for the trained Decision Trees. The *Title* is identified by all the methods as being the most discriminative field, but that is only because the *Abstract* is not present in all the pairs.

7.5.5 Crowdsourcing Deduplication

When using crowdsourcing platforms, the most important part is a very solid task description. Workers do not have the same background and can have very different understandings if they do not understand the task description perfectly. Only after having several versions and iterations of the task description, discussed with persons from different backgrounds we could gather better results.

Using qualification tests is only useful for some type of tasks. For us, for a relatively simple task, it drove most workers away. We imagine it makes more sense for tasks where workers have to exhibit specific skills – then the qualification test should be different than the actual HIT and test only these skills.

By comparing cases where all workers disagreed with our own judgment (this happened in 4% of the cases) we found out that most problems arise from workers being too quick and not paying enough attention to the task. Workers seem to take a quick look only at the beginning of metadata fields and not pay attention to the entire content. Such examples had even titles like: *"Proceedings of the 2003 ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '03): San Diego, California, USA, June 11 - 13, 2003"* and *"Proceedings of the 2003 ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation (PEPM '03): San Diego, California, USA, June 7th, 2003"* where all workers classified the publications as duplicates. In some other cases we could decide better only because our background and experience, e.g. two almost identical publications, one being a technical report and another one the phd thesis.

7.6 Conclusions

In this chapter we investigated a methodology for employing crowdsourcing in active learning, and identified some of the challenges that arise. We underline the importance of choosing an appropriate automatic method that learns from the crowd as this choice directly influences the selection of candidate instances.

In our investigations regarding the allocation of resources we have discovered evidence that there exists a certain threshold over which spending more money either in terms of the number of workers or the number of tasks per active learning round does not give a proportional increase in quality. The quality seems to rise as we allocate more resources, but after a certain point in seems to plateau. This is task-related and we do not claim that the limits we discovered in our experiments are universal. Carrying out such experiments when employing methods like this in order to determine the right level of resource allocation, therefore helping in keeping within the budget constraint are recommended. The selection strategy plays an important role in how fast the automatic method learns from the crowd. In our experiments we have noticed that a representative strategy is superior to the uncertain strategy.

By learning to approximate human decisions, we increase the duplicate detection accuracy up to 14%. In comparison, human assessors perform only 4% better than our automatic system. We show how to include such a duplicate detection module in a fully functional online publication search system where the focus is both on precision and on throughput.

The proposed framework for performing active learning with crowdsourcing is complex and has many components that rely on each other to function well. We have experimentally tested some of the components and their interplay in a well defined scenario, and we believe that the proposed methods can be successfully applied in other domains for other tasks similarly.

For future work we consider experiments using other types of tasks and data, using and comparing different label aggregation strategies. We intend to delve further into the influence of agreement on the quality of the labels and of the trained methods. We plan to look more closely into advanced worker quality assessments, taking into account the temporal dimension, and trying to recover those workers that consistently introduce errors. As a continuation to the task on which we experimentally apply our methods we could enhance the duplicates identification pipeline by to learning from the crowd how to create a unified representation of all the detected duplicates to be displayed in a search engine, in order not to clutter the results list.

A feedback mechanism for workers could be employed such that the automatic method can support the worker when he commits a mistake and attempt to correct him so that the humans and machines truly work together learning from one another. For this we would need worker authentication, in order to control the assignment process. The controlled assignment will allow us to use only the best workers or the workers that seem to learn and improve over time.

8 Conclusions and Outlook

This thesis investigates exploiting the wisdom of the crowds, and leveraging collective intelligence for well defined purposes. In particular, in the first part of the thesis we proposed to exploit the success of Wikipedia, to investigate individual contributions, and therefore identify information that is related to events. In the second part of the thesis we proposed methods for using crowdsourcing to improve machine learning, by tackling the issues that arise from the quality and unknown provenance of crowd-workers, and proposing a framework for active learning from the crowd.

8.1 Summary of Contributions

In the first part of thesis we investigated how collaborative intelligence, expressed as the reaction of Wikipedia contributors, can be exploited in order to identify events. In Chapter 2 we motivated our intuition for supporting the task of event detection on the exploitation of crowd intelligence as manifested in Wikipedia, and provided a review of the related literature. In Chapter 3 we analyzed how the community of contributors behind Wikipedia mobilizes itself and acts as an intelligent crowd when events happen that lead to the update of the articles of the involved entities. Based on these observations, we proposed methods for identifying the updates that were caused as an effect of the event, and furthermore for summarizing them. The proposed approach offers a comprehensive view over the event, discovering all the information that at the time was considered to be important by the crowd, but later on might have been removed or overly summarized, for the sake of brevity or to maintain the encyclopedic standards of Wikipedia. Moreover, we presented an online application that reports information about events where an entity was involved based on the proposed methods. In Chapter 4, we regard events as the interaction of entities with each other as either a cause or a consequence of a happening, and we present methods that leverage crowd wisdom can to detect events following this definition. Following the intuition that entities involved in the same event should be updated with similar

content, or even more specific, mentions of the other entities participating in one event we examined the concurrent updates that affect entities. Finally, we outlined the complementarity of our automatic approach of detecting events with a dedicated crowd contributed and curated Wikipedia portal about current events.

In the second part of the thesis we investigated a more direct way of exploiting the wisdom of the crowds, namely crowdsourcing. In Chapter 5 we introduced and motivated our approach for using crowdsourcing to improve machine learning, particularly for generating a high quality ground truth to be used in supervised machine learning algorithms, or by integrating crowdsourcing in an active learning framework where humans and machines collaborate, and provided a review of the related literature. In Chapter 6 we presented methods for the aggregation of multiple crowd provided labels to be used in machine learning, while tackling their noisy nature. The proposed methods simultaneously evaluate worker expertise and reliability, and find the underlying ground truth labels for a set of items. We evaluate our methods on various datasets proving their efficiency. In Chapter 7 we presented a framework for employing crowdsourcing for active learning, as an efficient way to gather of labeled instances. We investigated and tackled the challenges arisen by employing such a system such as the crowd label quality issue and the diverse resource allocation schemes and selection strategies. Furthermore, we integrated our proposed methods in a live publication search system.

8.2 Open Directions

In this thesis we presented ways of exploiting collective intelligence in order to tap into the wisdom of the crowds for improving event detection and machine learning. The methods proposed and the investigations conducted pave the way for future research directions.

Future extensions for exploiting the collaboration patterns in Wikipedia for event detection include studying opinions and controversies that occur in the context of event-related updates in Wikipedia. Uncovering this kind of information can be useful for providing users such as journalists and historians with more comprehensive overviews covering different schools of thought and points of view. More advanced linguistic and stylistic features of updates might be leveraged to improve classification and clustering. Moreover, updates and discussions can lead to further insights on social relationships between users, and provide clues about the provenance of event-related information contributed by different users. We can investigate who is running the discussions and what are the relationships between users working on the same articles, how to automatically extract a social network, or take into account the reputation of users. Another direction would be to evaluate the impact of the event. This can studied by examining the number of edits, views, links or citations, or of contributors. Impact can also be assessed by considering entries written in different

languages. This can shed more light on whether events are only of local importance or they have an international impact, and if different views or opinions may be noticed in different countries. Besides the general impact of the event, we can study the roles played by the participating entities, and their degree of involvement and importance and to which extent they were affected by the happening. The events that encompass the dynamic relationships discovered with our methods should be further summarized with a textual description to the event, to give a comprehensive view on the dynamics of the event and the involved entities. Another future extension can be the development of a probabilistic model for temporal retrieval and ranking, taking into account the temporal dimension, entity to event mappings, user involvement, as well as the dynamic multi-relational graphs from social networks.

Future directions for leveraging crowdsourcing towards improving machine learning can be envisioned both for the aggregation of crowd labels as well as for the extension of the active learning from crowds framework. The methods for aggregation of crowd labels can be extended by introducing different levels of supervision into the algorithms, or by simulating the behaviour of good users, based on the features of the tasks where they performed well. Therefore, even if workers that were considered to perform well do not participate in a task, we can replace them with an classifier that learned from the worker behavior how to perform the task in a similar way. In order to increase the quality of the crowd labels, we could aggregate only the labels from workers with experitse exceeding a threshold. Conversely, when evaluating the worker expertise we could only consider those instances for which the aggregated soft label exceeds a certain threshold. By updating the worker confidences online, an automatic mechanism that refuses labels from workers that were identified as being unreliable might benefit the requester of crowdsourced work, by keeping the costs low, and gathering just higher quality labels. As for active learning from the crowd, further experiments can be conducted by using other types of tasks and data, employing and comparing different label aggregation strategies. We intend to delve further into the influence of agreement on the quality of the labels and of the trained methods, and into advanced worker quality assessments, taking into account the temporal dimension, and trying to recover those workers that consistently introduce errors. The fatigue factor can also be considered, and when we notice that workers begin to underperform, we can suggest them to take a break for recovery. A truly symbiotic coupling of machine and human learning could be further facilitated by a feedback mechanism where the automatic method can support the worker when he commits a mistake and suggests a correction. Moreover, using worker profiles and controlled assignments would allow us to use only workers that perform well or those that seem to learn and improve over time. Furthermore, machine learning algorithms that take label uncertainty into acount could take advantage of the aggregated soft crowd label. Instance selection stragegies for label acquisition could be developed that take into account correlation between instances, agreement of the crowd on similar instances, the confidence of the automatic label, and the worker expertise in the same time.

A

Curriculum Vitae

Mihai Georgescu, born on November 27^{th}, 1984, in Rîmnicu Sărat, Romania.

2008 – present	**Ph.D. student** G. W. Leibniz Universität Hannover
2010 – present	**Junior Researcher** L3S Research Center, Hannover
2008 – 2010	**Graduate Research Assistant** L3S Research Center, Hannover
2003 – 2008	**Dipl.-Ing. in Computer Science** University "Politehnica", Faculty of Automatic Control and Computer Science, Bucharest

Bibliography

[ABK+07] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. *Dbpedia: A nucleus for a web of open data*. Springer, 2007.

[ABYBY11] Omar Alonso, Ricardo A. Baeza-Yates, and Ricardo A. Baeza-Yates. Design and implementation of relevance assessments using crowdsourcing. In *ECIR*, pages 153–164, 2011.

[ACdA+08] B. Thomas Adler, Krishnendu Chatterjee, Luca de Alfaro, Marco Faella, Ian Pye, and Vishwanath Raman. Assigning trust to Wikipedia content. In *Proceedings of WikiSym '08*, 2008.

[AdA07] B. Thomas Adler and Luca de Alfaro. A content-driven reputation system for the Wikipedia. In *Proceedings of WWW '07*, 2007.

[AGK10] Arvind Arasu, Michaela Götz, and Raghav Kaushik. On active learning of record matching packages. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 783–794. ACM, 2010.

[AHVC11] Vamshi Ambati, Sanjika Hewavitharana, Stephan Vogel, and Jaime Carbonell. Active learning with multiple annotations for comparable data classification task. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 69–77. Association for Computational Linguistics, 2011.

[All02] James Allan. *Topic detection and tracking: event-based information organization*, volume 12. Springer, 2002.

[AMM11] Eiji Aramaki, Sachiko Maskawa, and Mizuki Morita. Twitter catches the flu: Detecting influenza epidemics using Twitter. In *Proceedings of EMNLP '11*, 2011.

[AMN13] Omar Alonso, Catherine C Marshall, and Marc A Najork. A human-centered framework for ensuring reliability on crowdsourced labeling tasks. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[AMP06] Ofer Arazy, Wayne Morgan, and Raymond Patterson. Wisdom of the crowds: Decentralized knowledge construction in wikipedia. In *16th Annual Workshop on Information Technologies & Systems (WITS) Paper*, 2006.

[AMP10] Josh Attenberg, Prem Melville, and Foster Provost. Guided feature labeling for budget-sensitive learning under extreme class imbalance. In *ICML Workshop on Budgeted Learning*, 2010.

[AP10] Josh Attenberg and Foster Provost. Why label when you can search?: alternatives to active learning for applying human resources to build classification models under extreme class imbalance. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 423–432, New York, NY, USA, 2010. ACM.

[AP11] Josh Attenberg and Foster Provost. Inactive learning?: difficulties employing active learning in practice. *ACM SIGKDD Explorations Newsletter*, 12(2):36–41, 2011.

[APL98] James Allan, Ron Papka, and Victor Lavrenko. On-line new event detection and tracking. In *Proceedings of SIGIR '98*, 1998.

[ARM+11] Carlo Aliprandi, Francesco Ronzano, Andrea Marchetti, Maurizio Tesconi, and Salvatore Minutoli. Extracting events from wikipedia as rdf triples linked to widespread semantic web datasets. In *Online Communities and Social Computing*, pages 90–99. Springer, 2011.

[ASBYG11] Omar Alonso, Jannik Strötgen, Ricardo Baeza-Yates, and Michael Gertz. Temporal Information Retrieval: Challenges and Opportunities. In *WWW*, 2011.

[AVC10] Vamshi Ambati, Stephan Vogel, and Jaime G Carbonell. Active learning and crowd-sourcing for machine translation. In *LREC*, volume 11, pages 2169–2174. Citeseer, 2010.

[AVDCB11] Byung Gyu Ahn, Benjamin Van Durme, and Chris Callison-Burch. Wikitopics: what is popular on wikipedia and why. In *Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages*, pages 33–40. Association for Computational Linguistics, 2011.

[BAH12] Roja Bandari, Sitaram Asur, and Bernardo A. Huberman. The pulse of news in social media: Forecasting popularity. In *ICWSM*, 2012.

[BBC12] Alessandro Bozzon, Marco Brambilla, and Stefano Ceri. Answering search queries with crowdsearcher. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 1009–1018, New York, NY, USA, 2012. ACM.

[BCS+07] Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction for the web. In *IJCAI*, volume 7, pages 2670–2676, 2007.

[BCT07] Kurt Bollacker, Robert Cook, and Patrick Tufts. Freebase: A shared database of structured general human knowledge. In *AAAI*, volume 7, pages 1962–1963, 2007.

[BFG+09] Kerstin Bischoff, Claudiu S. Firan, Mihai Georgescu, Wolfgang Nejdl, and Raluca Paiu. Social knowledge-driven music hit prediction. In *ADMA*, pages 43–54, 2009.

[BFGM07] Abhijit Bhole, Blaž Fortuna, Marko Grobelnik, and Dunja Mladenić. Mining wikipedia and relating named entities over time. *People*, 100:62–4, 2007.

[BGC10] Anthony Brew, Derek Greene, and Pádraig Cunningham. The interaction between supervised learning and crowdsourcing. In *NIPS workshop on computational social science and the wisdom of crowds*, 2010.

[BGMG12] Yoram Bachrach, Thore Graepel, Tom Minka, and John Guiver. How to grade a test without knowing the answers—a bayesian graphical model for adaptive crowdsourcing and aptitude testing. *arXiv preprint arXiv:1206.6386*, 2012.

[BGMM+09] Omar Benjelloun, Hector Garcia-Molina, David Menestrina, Qi Su, Steven Euijong Whang, and Jennifer Widom. Swoosh: a generic approach to entity resolution. *The VLDB Journal*, 18(1):255–276, January 2009.

[BHK98] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the*

Fourteenth conference on Uncertainty in artificial intelligence, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[BIPR12] Kedar Bellare, Suresh Iyengar, Aditya G Parameswaran, and Vibhor Rastogi. Active sampling for entity matching. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1131–1139. ACM, 2012.

[BK13] Yukino Baba and Hisashi Kashima. Statistical quality estimation for general crowdsourcing tasks. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 554–562. ACM, 2013.

[BKLvR09] Ulrik Brandes, Patrick Kenis, Jürgen Lerner, and Denise van Raaij. Network analysis of collaboration structure in Wikipedia. In *Proceedings of WWW '09*, 2009.

[BL07] Ulrik Brandes and Jrgen Lerner. Revision and co-revision in Wikipedia. In *Bridging the Gap between Semantic Web and Web 2.0 (SemNet 2007)*, 2007.

[BL08] Ulrik Brandes and Jürgen Lerner. Visual analysis of controversy in user-generated encyclopedias. *Information Visualization*, 7(1):34–48, 2008.

[BMC+03] Mikhail Bilenko, Raymond Mooney, William Cohen, Pradeep Ravikumar, and Stephen Fienberg. Adaptive Name Matching in Information Integration. *IEEE Intelligent Systems*, 18(5):16–23, 2003.

[Bra08] Daren C Brabham. Crowdsourcing as a model for problem solving an introduction and cases. *Convergence: the international journal of research into new media technologies*, 14(1):75–90, 2008.

[CAL94] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Mach. Learn.*, 15(2):201–221, May 1994.

[CCH+08] David Crandall, Dan Cosley, Daniel Huttenlocher, Jon Kleinberg, and Siddharth Suri. Feedback effects between similarity and social influence in online communities. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 160–168. ACM, 2008.

[CGG+14] Andrea Ceroni, Mihai Georgescu, Ujwal Gadiraju, Kaweh Djafari Naini, and Marco Fisichella. Information evolution in wikipedia. In *Proceedings of The International Symposium on Open Collaboration*, page 24. ACM, 2014.

[CL11] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[CN10] Marek Ciglan and Kjetil Nørvåg. WikiPop: personalized event detection system based on Wikipedia page view statistics. In *Proceedings of CIKM '10*, 2010.

[CNKK14] Elisavet Chatzilari, Spiros Nikolopoulos, Yiannis Kompatsiaris, and Josef Kittler. Active learning in social context for image classification. In *9th International Conference on Computer Vision Theory and Applications, VISAPP*, 2014.

[CR09] Ling Chen and Abhishek Roy. Event detection from flickr data through wavelet-based spatial analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 523–532. ACM, 2009.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, 1995.

[DAGN12] Ernesto Diaz-Aviles, Mihai Georgescu, and Wolfgang Nejdl. Swarming to rank for recommender systems. In *RecSys*, pages 229–232, 2012.

[DAGSN10] Ernesto Diaz-Aviles, Mihai Georgescu, Avaré Stewart, and Wolfgang Nejdl. Lda for on-the-fly auto tagging. In *RecSys*, pages 309–312, 2010.

[DC08] Pinar Donmez and Jaime G Carbonell. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 619–628. ACM, 2008.

[DCS09] Pinar Donmez, Jaime G Carbonell, and Jeff Schneider. Efficiently learning the accuracy of labeling sources for selective sampling. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 259–268. ACM, 2009.

[DCS10] Pinar Donmez, Jaime G Carbonell, and Jeff G Schneider. A probabilistic framework to learn from multiple annotators with time-varying accuracy. In *SDM*, volume 2, page 1. SIAM, 2010.

[DDCM12] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 469–478, New York, NY, USA, 2012. ACM.

[DDCM13] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. Large-scale linked data integration using probabilistic reasoning and crowdsourcing. *The VLDB Journal*, 22(5):665–687, 2013.

[DDKR13] Nilesh Dalvi, Anirban Dasgupta, Ravi Kumar, and Vibhor Rastogi. Aggregating crowdsourced binary ratings. In *Proceedings of the 22nd international conference on World Wide Web*, pages 285–294. International World Wide Web Conferences Steering Committee, 2013.

[DFG+09] Gianluca Demartini, Claudiu S. Firan, Mihai Georgescu, Tereza Iofciu, Ralf Krestel, and Wolfgang Nejdl. An architecture for finding entities on the web. In *LA-WEB/CLIHC*, pages 230–237, 2009.

[DG13] Anirban Dasgupta and Arpita Ghosh. Crowdsourced judgement elicitation with endogenous proficiency. In *Proceedings of the 22nd international conference on World Wide Web*, pages 319–330. International World Wide Web Conferences Steering Committee, 2013.

[DLLH03] AnHai Doan, Ying Lu, Yoonkyong Lee, and Jiawei Han. Object Matching for Information Integration: A Profiler-Based Approach. In *IIWeb*, 2003.

[dMW10] Gerard de Melo and Gerhard Weikum. Menta: inducing multilingual taxonomies from wikipedia. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1099–1108. ACM, 2010.

[DRH11] Anhai Doan, Raghu Ramakrishnan, and Alon Y Halevy. Crowdsourcing systems on the world-wide web. *Communications of the ACM*, 54(4):86–96, 2011.

[DS79a] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics*, 28(1):20–28, 1979.

[DS79b] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, pages 20–28, 1979.

[DS09] Ofer Dekel and Ohad Shamir. Vox populi: Collecting high-quality labels from a crowd. 2009.

[DSJY11] Anish Das Sarma, Alpa Jain, and Cong Yu. Dynamic relationship and event discovery. In *WSDM*, 2011.

[EAGLdG12] Enrique Estellés-Arolas and Fernando González-Ladrón-de Guevara. Towards an integrated crowdsourcing definition. *Journal of Information science*, 38(2):189–200, 2012.

[ECD+05] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134, 2005.

[EG11] Miles Efron and Gene Golovchinsky. Estimation methods for ranking recent information. In *ACM SIGIR*, 2011.

[Faw06] Tom Fawcett. An introduction to ROC analysis. *Pattern Recogn. Lett.*, 27:861–874, June 2006.

[FBA10] Peter Kin-Fong Fong and Robert P Biuk-Aghai. What did they do? deriving high-level edit histories in wikis. In *Proceedings of the 6th International Symposium on Wikis and Open Collaboration*, page 2. ACM, 2010.

[Fer12] Michela Ferron. *Collective Memories in Wikipedia*. PhD thesis, University of Trento, 2012.

[FFK+11] Amber Feng, Michael J. Franklin, Donald Kossmann, Tim Kraska, Samuel Madden, Sukriti Ramesh, Andrew Wang, Reynold Xin, and Reynold Xin. Crowddb: Query processing with the vldb crowd. pages 1387–1390, 2011.

[FGN12] Claudiu S. Firan, Mihai Georgescu, and Wolfgang Nejdl. Social computing for libraries - data de-duplication through the crowd. In *10th International Bielefeld Conference, 2012, Bielefeld, Germany.*, 2012.

[FGNP10] Claudiu S Firan, Mihai Georgescu, Wolfgang Nejdl, and Raluca Paiu. Bringing order to your photos: event-driven classification of flickr images based on social knowledge. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 189–198. ACM, 2010.

[FGNS11] Claudiu S. Firan, Mihai Georgescu, Wolfgang Nejdl, and Xinyun Sun. Freesearch - literature search in a natural way. In *The Fifth Workshop on Human-Computer Interaction and Information Retrieval (HCIR), 2011, Mountain View, California, United States.*, 2011.

[FGNS12] Claudiu S. Firan, Mihai Georgescu, Wolfgang Nejdl, and Xinyun Sun. Freesearch: Literatursuche ohne hindernisse. In *Konferenz der Deutschen Gesellschaft für Informationswissenschaft und Informationspraxis (DGI) (poster paper), 2012, Dsseldorf, Germany.*, 2012.

[FKK+11] Michael J. Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, Reynold Xin, and Reynold Xin. Crowddb: answering queries with crowdsourcing. In *SIGMOD Conference*, pages 61–72, 2011.

[Fle05] Terry Flew. *New media: An introduction*. Oxford University Press, 2005.

[FLZZ11] Yifan Fu, Bin Li, Xingquan Zhu, and Chengqi Zhang. Do they belong to the same class: active learning by querying pairwise label homogeneity. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2161–2164. ACM, 2011.

[FM11a] Michela Ferron and Paolo Massa. The arab spring— wikirevolutions: Wikipedia as a lens for studying the real-time formation of collective memories of revolutions. *International Journal of Communication*, 5:20, 2011.

[FM11b] Michela Ferron and Paolo Massa. Collective memory building in wikipedia: the case of north african uprisings. In *Proceedings of the 7th International Symposium on Wikis and Open Collaboration*, pages 114–123. ACM, 2011.

[FM11c] Michela Ferron and Paolo Massa. Studying collective memories in wikipedia. *Journal of Social Theory*, 3(4):449–466, 2011.

[FM12] Michela Ferron and Paolo Massa. Psychological processes underlying wikipedia representations of natural and manmade disasters. In *Proceedings of the Eighth Annual International Symposium on Wikis and Open Collaboration*, page 2. ACM, 2012.

[FM13] Michela Ferron and Paolo Massa. Beyond the encyclopedia: Collective memories in wikipedia. *Memory Studies*, page 1750698013490590, 2013.

[FSDN10] Marco Fisichella, Avaré Stewart, Kerstin Denecke, and Wolfgang Nejdl. Unsupervised public health event detection for epidemic intelligence. In *Proceedings of CIKM '10*, 2010.

[FYYL05] Gabriel Pui Cheong Fung, Jeffrey X. Yu, Philip S. Yu, and Hongjun Lu. Parameter free bursty events detection in text streams. In *VLDB*, 2005.

[FZG11a] Oliver Ferschke, Torsten Zesch, and Iryna Gurevych. Wikipedia revision toolkit: efficiently accessing wikipedia's edit history. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations*, pages 97–102. Association for Computational Linguistics, 2011.

[FZG11b] Oliver Ferschke, Torsten Zesch, and Iryna Gurevych. Wikipedia revision toolkit: Efficiently accessing wikipedia's edit history. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. System Demonstrations*, pages 97–102, Portland, OR, USA, Jun 2011.

[FZL13] Yifan Fu, Xingquan Zhu, and Bin Li. A survey on instance selection for active learning. *Knowledge and information systems*, 35(2):249–283, 2013.

[Gal07] Francis Galton. Vox populi (the wisdom of crowds). *Nature*, 75, 1907.

[GAMS10] Alban Galland, Serge Abiteboul, Amélie Marian, and Pierre Senellart. Corroborating information from disagreeing views. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 131–140. ACM, 2010.

[Gil05] Jim Giles. Internet encyclopaedias go head to head. *Nature*, 438(7070):900–901, 2005.

[GK13] Anja Gruenheid and Donald Kossmann. Cost and quality trade-offs in crowdsourcing. *DBCrowd*, 1025:43–46, 2013.

[GKK+13] Mihai Georgescu, Nattiya Kanhabua, Daniel Krause, Wolfgang Nejdl, and Stefan Siersdorfer. Extracting Event-Related Information from Article Updates in Wikipedia. In *Advances in Information Retrieval - 35th European Conference on IR Research, ECIR 2013, Moscow, Russia, March 24-27, 2013. Proceedings*, pages 254–266, 2013.

[GKM11] Arpita Ghosh, Satyen Kale, and Preston McAfee. Who moderates the moderators?: crowdsourcing abuse detection in user-generated content. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 167–176. ACM, 2011.

[GM12] Arpita Ghosh and Preston McAfee. Crowdsourcing with endogenous entry. In *Proceedings of the 21st international conference on World Wide Web*, pages 999–1008. ACM, 2012.

[GNC+14] Ujwal Gadiraju, Kaweh Djafari Naini, Andrea Ceroni, Mihai Georgescu, Dang Duc Pham, and Marco Fisichella. WikipEvent: Temporal Event Data for the Semantic Web. In *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014.*, pages 125–128. CEUR-WS.org, 2014.

[GPF+12] Mihai Georgescu, Dang Duc Pham, Claudiu S. Firan, Wolfgang Nejdl, and Julien Gaugaz. Map to humans and reduce error: crowdsourcing for deduplication applied to digital libraries. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, pages 1970–1974, New York, NY, USA, 2012. ACM.

[GPF+14] Mihai Georgescu, Dang Duc Pham, Claudiu S Firan, Ujwal Gadiraju, and Wolfgang Nejdl. When in doubt ask the crowd: Employing crowdsourcing for active learning. In *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)*, page 12. ACM, 2014.

[GPGM12] Stephen Guo, Aditya Parameswaran, and Hector Garcia-Molina. So who won?: dynamic max discovery with the crowd. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 385–396. ACM, 2012.

[GPK+13] Mihai Georgescu, Dang Duc Pham, Nattiya Kanhabua, Sergej Zerr, Stefan Siersdorfer, and Wolfgang Nejdl. Temporal summarization of event-related updates in Wikipedia. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013, Companion Volume*, pages 281–284, 2013.

[GSD+12] Julien Gaugaz, Patrick Siehndel, Gianluca Demartini, Tereza Iofciu, Mihai Georgescu, and Nicola Henze. Predicting the future impact of news events. In *ECIR*, pages 50–62, 2012.

[GZ13] Mihai Georgescu and Xiaofei Zhu. L3S at MediaEval 2013 Crowdsourcing for Social Multimedia Task. In *Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop, Barcelona, Spain, October 18-19, 2013.* CEUR-WS.org, 2013.

[GZ14] Mihai Georgescu and Xiaofei Zhu. Aggregation of crowdsourced labels based on worker history. In *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)*, page 37. ACM, 2014.

[HCL07] Qi He, Kuiyu Chang, and Ee-Peng Lim. Analyzing feature trajectories for event detection. In *Proceedings of SIGIR '07*, 2007.

[HCMF+12] Mehdi Hosseini, Ingemar J Cox, Nataša Milić-Frayling, Gabriella Kazai, and Vishwa Vinay. On aggregating labels from multiple crowd workers to infer relevance of documents. In *Advances in Information Retrieval*, pages 182–194. Springer, 2012.

[HDM+05] Alon Y Halevy, Xin Dong, Jayant Madhavan, Alon Y Halevy, and Jayant Madhavan. Reference reconciliation in complex information spaces. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data - SIGMOD '05*, page 85, New York, New York, USA, 2005. ACM Press.

[HFH+09] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[HGH+12] James A Hammerton, Michael Granitzer, Dan Harvey, Maya Hristakeva, and Kris Jack. On generating large-scale ground truth datasets for the deduplication of bibliographic records. In *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, page 18. ACM, 2012.

[HL12] Daniel Hienert and Francesco Luciano. Extraction of historical events from wikipedia. *arXiv preprint arXiv:1205.4138*, 2012.

[HMS09] Pei-Yun Hsueh, Prem Melville, and Vikas Sindhwani. Data quality from crowdsourcing: a study of annotation selection criteria. In *Proceedings of the NAACL HLT 2009 workshop on active learning for natural language processing*, pages 27–35. Association for Computational Linguistics, 2009.

[How06] Jeff Howe. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.

[HS98] Mauricio A Hernández and Salvatore J Stolfo. Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem. *Data Min. Knowl. Discov.*, 2(1):9–37, 1998.

[HSBW12] Johannes Hoffart, Fabian Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence Journal, Special Issue on Wikipedia and Semi-Structured Resources*, 2012.

[INN08] Ekaterini Ioannou, Claudia Niederée, and Wolfgang Nejdl. Probabilistic Entity Linkage for Heterogeneous Information Spaces. In *CAiSE*, 2008.

[IP11] Panagiotis G Ipeirotis and Praveen K Paritosh. Managing crowdsourced human computation: a tutorial. In *Proceedings of the 20th international conference companion on World wide web*, pages 287–288. ACM, 2011.

[Ipe10] Panagiotis G Ipeirotis. Analyzing the amazon mechanical turk marketplace. *XRDS: Crossroads, The ACM Magazine for Students*, 17(2):16–21, 2010.

[IPW10] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67. ACM, 2010.

[JL11] Hyun Joon Jung and Matthew Lease. Improving consensus accuracy via z-score and weighted voting. In *Human Computation*, 2011.

[JL12a] Hyun Joon Jung and Matthew Lease. Improving quality of crowdsourced labels via probabilistic matrix factorization. In *Proceedings of the 4th Human Computation Workshop (HCOMP) at AAAI*, pages 101–106, 2012.

[JL12b] Hyun Joon Jung and Matthew Lease. Inferring missing relevance judgments from crowd workers via probabilistic matrix factorization. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1095–1096. ACM, 2012.

[JL12c] David Jurgens and Tsai-Ching Lu. Temporal motifs reveal the dynamics of editor interactions in wikipedia. In *ICWSM*, 2012.

[JLP+14] Tatiana Josephy, Matt Lease, Praveen Paritosh, Markus Krause, Mihai Georgescu, Michael Tjalve, and Daniela Braga. Workshops held at the first AAAI conference on human computation and crowdsourcing: A report. *AI Magazine*, 35(2):75–78, 2014.

[JSL09] Rut Jesus, Martin Schwartz, and Sune Lehmann. Bipartite networks of wikipedia's articles and authors: a meso-level approach. In *Proceedings of the 5th International Symposium on Wikis and Open Collaboration*, page 5. ACM, 2009.

[Jun14] Hyun Joon Jung. Quality assurance in crowdsourcing via matrix factorization based task routing. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 3–8. International World Wide Web Conferences Steering Committee, 2014.

[KBG13] Markus Krause, François Bry, and Mihai Georgescu. Disco: Workshop on human and machine learning in games. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[KBK+12] Michal Kosinski, Yoram Bachrach, Gjergji Kasneci, Jurgen Van-Gael, and Thore Graepel. Crowd iq: Measuring the intelligence of crowdsourcing platforms. In *Proceedings of the 3rd Annual ACM Web Science Conference*, pages 151–160. ACM, 2012.

[KCP+07] Aniket Kittur, Ed Chi, Bryan A Pendleton, Bongwon Suh, and Todd Mytkowicz. Power of the few vs. wisdom of the crowd: Wikipedia and the rise of the bourgeoisie. *World wide web*, 1(2):19, 2007.

[KGC11] Brian Keegan, Darren Gergle, and Noshir Contractor. Hot off the wiki: dynamics, practices, and structures in wikipedia's coverage of the tōhoku catastrophes. In *Proceedings of the 7th International Symposium on Wikis and Open Collaboration*, pages 105–113. ACM, 2011.

[KGC12] Brian Keegan, Darren Gergle, and Noshir Contractor. Staying in the loop: Structure and dynamics of wikipedia's breaking news collaborations. In *Proceedings of WikiSym '12*, 2012.

[KGSG11] Gjergji Kasneci, Jurgen Van Gael, David Stern, and Thore Graepel. CoBayes: bayesian knowledge corroboration with assessors of unknown areas of expertise. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 465–474. ACM, 2011.

[KHH12] Ece Kamar, Severin Hacker, and Eric Horvitz. Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 467–474. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[KK08] Aniket Kittur and Robert E Kraut. Harnessing the wisdom of crowds in wikipedia: quality through coordination. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 37–46. ACM, 2008.

[KKK+11] Gabriella Kazai, Jaap Kamps, Marijn Koolen, Natasa Milic-Frayling, and Natasa Milic-Frayling. Crowdsourcing for book search evaluation: impact of hit design on comparative system ranking. In *SIGIR*, pages 205–214, 2011.

[KL11] Abhimanu Kumar and Matthew Lease. Modeling annotator accuracies for supervised learning. In *WSDM Workshop on Crowdsourcing for Search and Data Mining*, pages 19–22, 2011.

[Kle02] Jon Kleinberg. Bursty and hierarchical structure in streams. KDD '02, pages 91–101, New York, NY, USA, 2002. ACM.

[KN12] Nattiya Kanhabua and Kjetil Nørvåg. Learning to rank search results for time-sensitive queries. In *CIKM*, 2012.

[KNB+13] Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton.

The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 1301–1318. ACM, 2013.

[KOS11] David R Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems*, pages 1953–1961, 2011.

[KSKK11] Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E Kraut. Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 43–52. ACM, 2011.

[KSPC07] Aniket Kittur, Bongwon Suh, Bryan A. Pendleton, and Ed H. Chi. He says, she says: conflict and coordination in Wikipedia. In *Proceedings of CHI '07*, 2007.

[KTK12] Hiroshi Kajino, Yuta Tsuboi, and Hisashi Kashima. A convex formulation for learning from crowds. In *AAAI*, 2012.

[KVGHG10] Gjergji Kasneci, Jurgen Van Gael, Ralf Herbrich, and Thore Graepel. Bayesian knowledge corroboration with logical rules and user feedback. In *Machine Learning and Knowledge Discovery in Databases*, pages 1–18. Springer, 2010.

[KW12a] Erdal Kuzey and Gerhard Weikum. Extraction of temporal facts and events from Wikipedia. In *Proceedings of the 2nd Temporal Web Analytics Workshop*, TempWeb '12, 2012.

[KW12b] Erdal Kuzey and Gerhard Weikum. Extraction of temporal facts and events from wikipedia. TempWeb '12. ACM, 2012.

[LB99] Pierre Lévy and Robert Bonomo. *Collective intelligence: Mankind's emerging world in cyberspace*. Perseus Publishing, 1999.

[LCR+14] Babak Loni, Lei Yen Cheung, Michael Riegler, Alessandro Bozzon, Martha Larson, and Luke Gottlieb. Fashion 10000: An enriched social image dataset for fashion and clothing. In *Proceedings of the 5th ACM Multimedia Systems Conference*, MMSys '14, 2014.

[LDLD08] Minh-Tam Le, Hoang-Vu Dang, Ee-Peng Lim, and Anwitaman Datta. Wikinetviz: Visualizing friends and adversaries in implicit social networks. In *Intelligence and Security Informatics, 2008. ISI 2008. IEEE International Conference on*, pages 52–57. IEEE, 2008.

[LDS12] Chenliang Li, Anwitaman Datta, and Aixin Sun. Mining latent relations in peer-production environments: a case study with wikipedia

article similarity and controversy. *Social Network Analysis and Mining*, 2(3):265–278, 2012.

[Lea11] Matthew Lease. On quality control and machine learning in crowdsourcing. In *Human Computation*, 2011.

[LGRT11] Martin Lukasiewycz, Michael Glaß, Felix Reimann, and Jürgen Teich. Opt4J - A Modular Framework for Meta-heuristic Optimization. In *Proceedings of the Genetic and Evolutionary Computing Conference (GECCO 2011)*, Dublin, Ireland, 2011.

[LHG+14] Babak Loni, Jonathon Hare, Mihai Georgescu, Michael Riegler, Xiaofei Zhu, Mohamed Morchid, Richard Dufour, and Martha Larson. Getting by with a little help from the crowd: Practical approaches to social image labeling. *CROWDMM*, 14:03–07, 2014.

[Lih04] Andrew Lih. Wikipedia as participatory journalism: Reliable sources? metrics for evaluating collaborative media as a news resource. In *the 5th International Symposium on Online Journalism*, 2004.

[LL09] Michael D Lieberman and Jimmy Lin. You are where you edit: Locating wikipedia contributors through edit histories. In *ICWSM*, 2009.

[LLBG13] Babak Loni, Martha Larson, Alessandro Bozzon, and Luke Gottlieb. Crowdsourcing for social multimedia at mediaeval 2013: Challenges, data set, and evaluation. In *MediaEval 2013 Workshop, Barcelona, Spain*, 2013.

[LMG+13] Babak Loni, Maria Menendez, Mihai Georgescu, Luca Galli, Claudio Massari, Ismail Sengör Altingövde, Davide Martinenghi, Mark Melenhorst, Raynor Vliegendhart, and Martha Larson. Fashion-focused creative commons social dataset. In *Proceedings of the 4th ACM Multimedia Systems Conference*, pages 72–77. ACM, 2013.

[LSS11] Florian Laws, Christian Scheible, and Hinrich Schütze. Active learning with amazon mechanical turk. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1546–1556. Association for Computational Linguistics, 2011.

[LVA09] Edith Law and Luis Von Ahn. Input-agreement: a new mechanism for collecting data using human computation games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1197–1206. ACM, 2009.

[LWLM05] Zhiwei Li, Bin Wang, Mingjing Li, and Wei-Ying Ma. A probabilistic model for retrospective news event detection. In *Proceedings of SIGIR '05*, 2005.

[LYZ13] Hongwei Li, Bin Yu, and Dengyong Zhou. Error rate analysis of labeling by crowdsourcing. In *ICML Workshop: Machine Learning Meets Crowdsourcing. Atalanta, Georgia, USA*, 2013.

[Mas11] Paolo Massa. Social networks of wikipedia. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, pages 221–230. ACM, 2011.

[MBB+10] Zoltán Miklós, Nicolas Bonvin, Paolo Bouquet, Michele Catasta, Daniele Cordioli, Peter Fankhauser, Julien Gaugaz, Ekaterini Ioannou, Hristo Koshutanski, Antonio Maña, Claudia Niederée, Themis Palpanas, and Heiko Stoermer. From Web Data to Entities and Back. *CAiSE*, pages 302–316, June 2010.

[MBKK13] Toshiko Matsui, Yukino Baba, Toshihiro Kamishima, and Hisashi Kashima. Crowdsourcing quality control for item ordering tasks. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[MKM+12] Adam Marcus, David Karger, Samuel Madden, Robert Miller, and Sewoong Oh. Counting with the crowd. *Proceedings of the VLDB Endowment*, 6(2):109–120, 2012.

[MMLW09] Olena Medelyan, David Milne, Catherine Legg, and Ian H. Witten. Mining meaning from Wikipedia. *Int. J. Hum.-Comput. Stud.*, 67:716–754, September 2009.

[MS99] C. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT Press, 1999.

[MSF+12] Barzan Mozafari, Purnamrita Sarkar, Michael J Franklin, Michael I Jordan, and Samuel Madden. Active learning for crowd-sourced databases. *arXiv preprint arXiv:1209.3686*, 2012.

[MVB08] A Morris, Y Velegrakis, and P Bouquet. Entity Identification on the Semantic Web. In *SWAP*, 2008.

[MW13] David Milne and Ian H Witten. An open-source toolkit for mining wikipedia. *Artificial Intelligence*, 194:222–239, 2013.

[MWK+11a] Adam Marcus, Eugene Wu, David Karger, Samuel Madden, and Robert Miller. Human-powered sorts and joins. *Proceedings of the VLDB Endowment*, 5(1):13–24, 2011.

[MWK+11b] Adam Marcus, Eugene Wu, David R Karger, Samuel Madden, and Robert C Miller. Demonstration of qurk: a query processor for human-operators. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 1315–1318. ACM, 2011.

[NP12] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

[NRD08] Sérgio Nunes, Cristina Ribeiro, and Gabriel David. WikiChanges: exposing Wikipedia revision activity. In *Proceedings of WikiSym '08*, 2008.

[OGB07] Felipe Ortega and Jesus M Gonzalez Barahona. Quantitative analysis of thewikipedia community of users. In *Proceedings of the 2007 international symposium on Wikis*, pages 75–86. ACM, 2007.

[OK08] Harri Oinas-Kukkonen. Network analysis and crowds of people as sources of new organisational knowledge. *Knowledge Management: Theoretical Foundation. Informing Science Press, Santa Rosa, CA, US*, pages 173–189, 2008.

[Ols09] Fredrik Olsson. A literature survey of active machine learning in the context of natural language processing. 2009.

[OPM+12] Miles Osborne, Saša Petrovic, Richard McCreadie, Craig Macdonald, and Iadh Ounis. Bieber no more: First story detection using twitter and wikipedia. In *Proceedings of the Workshop on Time-aware Information Access. TAIA*, volume 12, 2012.

[Pen09] Christian Pentzold. Fixing the floating gap: The online encyclopaedia wikipedia as a global memory place. *Memory Studies*, 2(2):255–272, 2009.

[PGMP+12] Hyunjung Park, Hector Garcia-Molina, Richard Pang, Neoklis Polyzotis, Aditya Parameswaran, and Jennifer Widom. Deco: A system for declarative crowdsourcing. *Proceedings of the VLDB Endowment*, 5(12):1990–1993, 2012.

[PMdR] Maria-Hendrike Peetz, Edgar Meij, and Maarten de Rijke. Opengeist: Insight in the stream of page views on wikipedia. In *SIGIR 2012 Workshop on Time-aware Information Access, TAIA*, volume 12.

[PN09] Simone Paolo Ponzetto and Roberto Navigli. Large-scale taxonomy mapping for restructuring and integrating wikipedia. In *IJCAI*, volume 9, pages 2083–2088, 2009.

[PS11] Simone Paolo Ponzetto and Michael Strube. Taxonomy induction based on a collaboratively built knowledge repository. *Artificial Intelligence*, 175(9-10):1737–1756, 2011.

[QB09] Alexander J Quinn and Benjamin B Bederson. A taxonomy of distributed human computation. *Human-Computer Interaction Lab Tech Report, University of Maryland*, 2009.

[QB11] Alexander J Quinn and Benjamin B Bederson. Human computation: a survey and taxonomy of a growing field. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1403–1412. ACM, 2011.

[RGN07] Tye Rattenbury, Nathaniel Good, and Mor Naaman. Towards automatic extraction of event and place semantics from flickr tags. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 103–110. ACM, 2007.

[RYZ+10] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *The Journal of Machine Learning Research*, 99:1297–1322, 2010.

[Sab07] Mikalai Sabel. Structuring wiki revision history. In *Proceedings of the 2007 international symposium on Wikis*, pages 125–130. ACM, 2007.

[SB02] Sunita Sarawagi and Anuradha Bhamidipaty. Interactive deduplication using active learning. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–278. ACM, 2002.

[SBF+09] Ruben Sipoš, Abhijit Bhole, Blaž Fortuna, Marko Grobelnik, and Dunja Mladenić. Demo: Historyviz–visualizing events and relations extracted from wikipedia. In *The Semantic Web: Research and Applications*, pages 903–907. Springer, 2009.

[SBKM02] Sunita Sarawagi, Anuradha Bhamidipaty, Alok Kirpal, and Chandra Mouli. Alias: An active learning led interactive deduplication system. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 1103–1106. VLDB Endowment, 2002.

[SCF08] Burr Settles, Mark Craven, and Lewis Friedland. Active learning with real annotation costs. In *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*, pages 1–10, 2008.

[SCKP08] Bongwon Suh, Ed H Chi, Aniket Kittur, and Bryan A Pendleton. Lifting the veil: improving accountability and social transparency in wikipedia with wikidashboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1037–1040. ACM, 2008.

[SCPK07] Bongwon Suh, Ed H Chi, Bryan A Pendleton, and Aniket Kittur. Us vs. them: Understanding social dynamics in wikipedia with revert graph visualizations. In *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on*, pages 163–170. IEEE, 2007.

[Set10] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 2010.

[SH07] Klaus Stein and Claudia Hess. Does it matter who contributes: a study on featured articles in the German Wikipedia. In *Proceedings of HT '07*, 2007.

[SKW07] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.

[SKW08] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A large ontology from wikipedia and wordnet. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3):203–217, 2008.

[SL13a] Aashish Sheshadri and Matthew Lease. SQUARE: A Benchmark for Research on Computing Crowd Consensus. In *Proceedings of the 1st AAAI Conference on Human Computation (HCOMP)*, pages 156–164, 2013.

[SL13b] Aashish Sheshadri and Matthew Lease. SQUARE: Benchmarking Crowd Consensus at MediaEval. In *Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop*, 2013. CEUR Workshop (ceur-ws.org) Proceedings Vol-1043, ISSN 1613-0073.

[SOJN08] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263. Association for Computational Linguistics, 2008.

[SOM10] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of WWW '10*, 2010.

[SPI08] Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622. ACM, 2008.

[SRMRB12] Hoda Sepehri Rad, Aibek Makazhanov, Davood Rafiei, and Denilson Barbosa. Leveraging editor collaboration patterns in wikipedia. In *Proceedings of the 23rd ACM conference on Hypertext and social media*, pages 13–22. ACM, 2012.

[Sur05] James Surowiecki. *The wisdom of crowds*. Random House LLC, 2005.

[SVHS13] Thomas Steiner, Seth Van Hooland, and Ed Summers. Mj no more: using concurrent wikipedia edit spikes with social network plausibility checks for breaking news detection. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 791–794. International World Wide Web Conferences Steering Committee, 2013.

[TA14] Giang Binh Tran and Mohammad Alrifai. Indexing and analyzing wikipedia's current events portal, the daily news summaries by the crowd. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 511–516. International World Wide Web Conferences Steering Committee, 2014.

[TCG+14] Tuan Tran, Andrea Ceroni, Mihai Georgescu, Kaweh Djafari Naini, and Marco Fisichella. Wikipevent: Leveraging wikipedia edit history for event detection. In *Web Information Systems Engineering–WISE 2014*, pages 90–108. Springer, 2014.

[TEPW11] Tran Anh Tuan, Shady Elbassuoni, Nicoleta Preda, and Gerhard Weikum. Cate: context-aware timeline for entity illustration. WWW '11. ACM, 2011.

[TGZK14] Tuan Tran, Mihai Georgescu, Xiaofei Zhu, and Nattiya Kanhabua. Analysing the duration of trending topics in twitter using wikipedia. In *Proceedings of the 2014 ACM conference on Web science*, pages 251–252. ACM, 2014.

[TK01] Simon Tong and Daphne Koller. Active learning for structure in bayesian networks. In *International joint conference on artificial intelligence*, volume 17, pages 863–869. LAWRENCE ERLBAUM ASSOCIATES LTD, 2001.

[TL11] Wei Tang and Matthew Lease. Semi-supervised consensus labeling for crowdsourcing. In *SIGIR 2011 Workshop on Crowdsourcing for Information Retrieval (CIR)*, 2011.

[TLB+11] Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, Adam Kalai, and Adam Kalai. Adaptively learning the crowd kernel. In *ICML*, pages 673–680, 2011.

[TOY+11] Yuku Takahashi, Hiroaki Ohshima, Mitsuo Yamamoto, Hirotoshi Iwasaki, Satoshi Oyama, and Katsumi Tanaka. Evaluating significance of historical entities based on tempo-spatial impacts analysis using wikipedia link structure. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, pages 83–92. ACM, 2011.

[VA06] Luis Von Ahn. Games with a purpose. *Computer*, 39(6):92–94, 2006.

[VA09a] Luis Von Ahn. Human computation. In *Design Automation Conference, 2009. DAC'09. 46th ACM/IEEE*, pages 418–419. IEEE, 2009.

[vA09b] Luis von Ahn. Human computation. In *CIVR*, 2009.

[VG14] Sudheendra Vijayanarasimhan and Kristen Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. *International Journal of Computer Vision*, 108(1-2):97–114, 2014.

[VGK+14] Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd international conference on World wide web*, pages 155–164. International World Wide Web Conferences Steering Committee, 2014.

[VGM12] Petros Venetis and Hector Garcia-Molina. Quality control for comparison microtasks. In *Proceedings of the First International Workshop on Crowdsourcing and Data Mining*, pages 15–21. ACM, 2012.

[VGMH+12] Petros Venetis, Hector Garcia-Molina, Kerui Huang, Neoklis Polyzotis, and Neoklis Polyzotis. Max algorithms in crowdsourcing environments. In *WWW*, pages 989–998, 2012.

[VJG10] Sudheendra Vijayanarasimhan, Prateek Jain, and Kristen Grauman. Far-sighted active learning on a budget for image and video recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3035–3042. IEEE, 2010.

[VLS+08] Ba-Quy Vuong, Ee-Peng Lim, Aixin Sun, Minh-Tam Le, and Hady Wirawan Lauw. On ranking controversies in Wikipedia: models and evaluation. In *Proceedings of WSDM '08*, 2008.

[VRJ13] Matteo Venanzi, Alex Rogers, and Nicholas R Jennings. Trust-based fusion of untrustworthy information in crowdsourcing applications. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 829–836. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[VWD04] Fernanda B Viégas, Martin Wattenberg, and Kushal Dave. Studying cooperation and conflict between authors with history flow visualizations. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 575–582. ACM, 2004.

[WBPB10] Peter Welinder, Steve Branson, Pietro Perona, and Serge J Belongie. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems*, pages 2424–2432, 2010.

[WH11] Meng Wang and Xian-Sheng Hua. Active learning in multimedia annotation and retrieval: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(2):10, 2011.

[WJA14] Stewart Whiting, Joemon Jose, and Omar Alonso. Wikipedia as a time machine. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 857–862. International World Wide Web Conferences Steering Committee, 2014.

[WKFF12] Jiannan Wang, Tim Kraska, Michael J Franklin, and Jianhua Feng. Crowder: Crowdsourcing entity resolution. *Proceedings of the VLDB Endowment*, 5(11):1483–1494, 2012.

[WP09] Thomas Wöhner and Ralf Peters. Assessing the quality of wikipedia articles with lifecycle based metrics. In *Proceedings of the 5th International Symposium on Wikis and Open Collaboration*, page 16. ACM, 2009.

[WP10] Peter Welinder and Pietro Perona. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 25–32. IEEE, 2010.

[WSBT11] Byron C Wallace, Kevin Small, Carla E Brodley, and Thomas A Trikalinos. Who should label what? instance allocation in multiple expert active learning. In *SDM*, pages 176–187. SIAM, 2011.

[WWA+08] Daniel S Weld, Fei Wu, Eytan Adar, Saleema Amershi, James Fogarty, Raphael Hoffmann, Kayur Patel, and Michael Skinner. Intelligence in wikipedia. In *AAAI*, volume 8, pages 1609–1614, 2008.

[WWB+09] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*, pages 2035–2043, 2009.

[WZQ+10] Yafang Wang, Mingjie Zhu, Lizhen Qu, Marc Spaniol, and Gerhard Weikum. Timely yago: harvesting, querying, and visualizing temporal knowledge from wikipedia. In *Proceedings of the 13th International Conference on Extending Database Technology*, pages 697–700. ACM, 2010.

[YCKK09] Man-Ching Yuen, Ling-Jyh Chen, Irwin King, and Irwin King. A survey of human computation systems. In *CSE (4)*, pages 723–728, 2009.

[YFRD11] Yan Yan, Glenn M Fung, Rómer Rosales, and Jennifer G Dy. Active learning from crowds. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1161–1168, 2011.

[YMSM10] Hui Yang, Anton Mityagin, Krysta M Svore, and Sergey Markov. Collecting high quality overlapping labels at low cost. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 459–466. ACM, 2010.

[YPC98] Yiming Yang, Tom Pierce, and Jaime Carbonell. A study of retrospective and on-line event detection. In *ACM SIGIR*, 1998.

[ZBMP12] Dengyong Zhou, Sumit Basu, Yi Mao, and John C Platt. Learning from the wisdom of crowds by minimax entropy. In *Advances in Neural Information Processing Systems*, pages 2195–2203, 2012.

[ZCB11] Omar F Zaidan and Chris Callison-Burch. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1220–1229. Association for Computational Linguistics, 2011.

[ZLBM06] Qiankun Zhao, Tie-Yan Liu, Sourav S Bhowmick, and Wei-Ying Ma. Event detection from evolution of click-through data. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 484–493. ACM, 2006.

[ZMG08] Torsten Zesch, Christof Müller, and Iryna Gurevych. Extracting lexical semantic knowledge from wikipedia and wiktionary. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, Marrakech, Morocco, May 2008. electronic proceedings.

[ZNG14] Xiaofei Zhu, Wolfgang Nejdl, and Mihai Georgescu. An adaptive teleportation random walk model for learning social tag relevance. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 223–232. ACM, 2014.

[ZS03] Yunyue Zhu and Dennis Shasha. Efficient elastic burst detection in data streams. In *Proceedings of KDD '03*, 2003.

[ZSS11] Liyue Zhao, Gita Sukthankar, and Rahul Sukthankar. Incremental relabeling for active learning with noisy crowdsourced annotations. In *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*, pages 728–733. IEEE, 2011.

[ZZS14] Liyue Zhao, Yu Zhang, and Gita Sukthankar. An active learning approach for jointly estimating worker performance and annotation reliability with crowdsourced data. *arXiv preprint arXiv:1401.3836*, 2014.