Entities on the Web: Resolution, Matching and Profiling

THÈSE N^{O} 5897 (2013)

PRÉSENTÉE LE 22 AOUT 2013

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE SYSTÈMES D'INFORMATION RÉPARTIS
PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Surender Reddy YERVA

acceptée sur proposition du jury:

Prof. M. Grossglauser, président du jury
Prof. K. Aberer, directeur de thèse
Dr K. Balog, rapporteur
Dr Z. Miklós, rapporteur
Dr M. Rajman, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2013

To my dear parents & family members.

Acknowledgements

This PhD thesis work would not have been possible if not for number of special people. I take this opportunity to thank all of them. First and foremost, I would like to express my deepest gratitude to my thesis adviser, Prof. Karl Aberer, for his invaluable guidance, insightful discussions, continuous support, and most importantly the creative freedom offered for carrying out thesis work on interesting research topics.

Secondly, I would like to thank Dr. Martin Rajman, Dr. Krisztian Balog, Dr. Zoltan Miklos, and Dr. Matthias Grossglauser for agreeing to be part of my thesis committee, and for providing valuable comments and suggestions for improving the thesis work.

My gratitude also goes to all my EPFL research collaborators without whom some of the chapters of this thesis could not have been possible. My sincere appreciation goes to Dr. Zoltan Miklos for guiding me throughout the thesis work, right from the initial papers to the very end of the thesis work. Special thanks to Michele Catasta, the big data geek, for all the collaborations and helping me throughout with the data collection and management. I would also like to thank Hoyoung, Wojtek, Dipanjan, and Luke Wright for the interesting discussions related to Twitter data mining. During the course of my PhD work, I appreciate having a chance to work with number of talented MS students: Flavia and Alex for the wonderful demos (TweetSpector & TripEneer) we built together, Adela Maznikar for decoding the URL spreading in Twitter Data, and others.

Numerous thanks to the wonderful LSIR lab colleagues (both the alumni and the current members), who made my stay here such a rewarding experience. Rammohan and Saket deserve a special mention. Thank you guys for always being there, be it serious technical discussions, pointless political debates, proofreading the drafts, or planning fun activities. Thank you Nicolas, Mehdi, Thanasis, Dipanjan, & others for the fun football sessions; Alexandra, Tri, Julien, Hao, and all the new members for organizing the social gatherings; Julien, Nicolas, Saket for helping me out with *French-to-English* translation on number of occasions. A special thanks to Chantal for all the help and support throughout my stay here.

I would also like to thank number of friends, co-travelers in the amazing *Swiss-life-boat* ride, for making life exciting beyond the thesis work. It was such a fun experience with the YUVA (Indian Students Association) group: Srini, Abhishek, Nishanth, Aamani, Nicee, Czuee, Shashi, Ganga, Soni, Payal, Nitesh, Maneesha, and many more. Thanks guys. It was always a pleasure being part of organizing the cultural activities (Yuvali), cricket sessions, skiing sessions or other fun activities with you folks. I would miss you guys and all the fun. The badminton group (Peru, Viswa, Aravind, Nachi, Saket, Devika, Mark, . . .) will always remain special to me. Thanks to friends back home (Seenu, Madhan, Nattu, Syntax, . . .) for always being there and the encouragement.

I would also like to thank the Vienet Family (Didier, Anne, Ambre, Romain) for giving me the flavor of Swiss Family experience. I am indebted to them for the little French I managed to learn here. Merci Beacoup.

Finally and most importantly, I would like to thank my parents and family members for their unconditional love and support, and for always encouraging me to pursue my dreams.

Surender Reddy Yerva
Lausanne, 6th *August, 2013*

Abstract

The majority of the information on the web is encoded as web documents in natural language for human consumption. According to International Data Corporation (IDC) 80% of the data on the web is unstructured (free text) and is growing at a rapid pace due to the ease with which data can be published on the blogs, social networks, web, etc. The fundamental idea of Semantic Web is to link all the knowledge on the web. For Semantic Web to be widely adopted, and to exploit its full potential, it is important that the researched techniques understand and automatically extract knowledge from the unstructured web documents, as majority of data on the web is unstructured. A promising approach to have programmatic access to such knowledge is the use of information extraction techniques. Most frequently these techniques aim at extracting entities, such as persons, geographic locations, etc., from free text. These entities can potentially be linked to each other, thus creating a de-facto global knowledge graph of linked entities. A number of entity-related challenges need to be addressed for realizing the entity-oriented view of Semantic Web. In this doctoral thesis, we provide research contributions to the field of entity extraction from Web text documents with the aim of facilitating the adoption of the Semantic Web. This thesis addresses following entity-related problems: *Entity Resolution for web documents*; *Entity Matching in micro-blogging environments*; and *Entity Profiling and Applications*. More specifically, we make the following contributions:

(1) ***Entity Resolution for Web Documents:*** One of the key challenges to realize automated processing of the information on the Web is related to the entity resolution problem. There are a number of tools that reliably recognize named entities, such as persons, companies, geographic locations, in Web documents. The names of these extracted entities are however non-unique; the same name on different Web pages might or might not refer to the same entity. We address this disambiguation problem, which is very similar to the entity resolution problem studied in relational databases, however there are also several differences. Most importantly Web pages often only contain partial or incomplete information about the entities. We propose a generic framework where multiple similarity functions corresponding to the domain specific rules can be defined. We make use of techniques from graph theory and machine learning for efficiently combining the evidence from multiple similarity functions for improved ER results, and demonstrate the efficiency of our framework on two real-world datasets.

(2) ***Entities in micro-blogs like Twitter:*** Twitter is a popular micro-blogging service on the Web, where people can publish short messages, which then become visible to other users of the service. While the topics of these messages vary, there are a lot of messages where the users express their opinions about companies or their products. These messages are a rich source of information for companies for sentiment analysis or opinion

mining. There is however a great obstacle for analyzing the messages directly: as the company names are often ambiguous (e.g. apple, the fruit vs. Apple Inc.), one needs first to identify which messages are related to the company. We first present simple techniques that make use of company profiles, which we created semi-automatically from external Web sources. Our advanced techniques take ambiguity estimations into account and also automatically extend the basic company profiles through active learning from the Twitter stream itself. We demonstrate the effectiveness of our methods through an extensive set of experiments. We also present *TweetSpector* as a working prototype for entity-based classification of tweets.

(3) ***Entity Profiling and Applications:*** Entity profiling is the problem of constructing a compact representation (profile) of an entity, which summarizes the various mentions of an entity. We focus on constructing entity profiles to user and location entities, and show applications that make use of such entity profiles.

  (a) ***User-Entity Profiles on Social Networks:*** Users through their activities on social networks leave traces of their personalities. With the advances of content mining and modeling techniques, it should be possible to profile an user entity from his social network content. In this work we explore various techniques for summarizing a user's presence on different social networks. We show that one of the advantages of maintaining user profile is to provide the context for understanding the short texts, and help in better understanding of microposts. Additionally, we present *TripEneer*: Travel plan recommendation application based on user and location entity profiles.

  (b) ***Social and Sensor Data Fusion of a Location-entity in the Cloud:*** As mobile cloud computing facilitates a wide spectrum of smart applications, the need for fusing various types of data available in the cloud grows rapidly. In particular, social and sensor data lies at the core in such applications, but is typically processed separately. This work explores the potential of fusing social and sensor data, related to a location entity, in the cloud. We present a travel recommendation system that is built upon a conceptual framework. This framework allows to blend the heterogeneous social and sensor data for integrated analysis, extracting weather-dependent people's mood information from Twitter and meteorological sensor data streams.

This thesis through these contributions for linking entities on the web makes a promising step towards realizing entity-oriented view of (Semantic) Web.

Keywords: *entity*, *entity resolution*, *entity profiles*, *recommendations*, *semantic web*, *twitter streams*, *social networks*, *machine learning*, *online reputation management*, *sentiment metrics*, *user models*

Résumé

La majorité de l'information sur le web est codée sous forme de documents web en langage courant pour la consommation par l'homme. Selon l'International Data Corporation (IDC), 80% des données sur le web ne sont pas structurées (en texte libre) et se développent à un rythme rapide en raison de la facilité avec laquelle les données peuvent être publiées sur les blogs, réseaux sociaux, web, etc. L'idée fondamentale du Web sémantique est de relier toutes les connaissances sur le web. Pour que le Web sémantique soit largement adopté, et exploité à son plein potentiel, il est important que les techniques, objets de cette recherche, comprennent et extraient automatiquement les connaissances à partir des documents Web non structurées, étant donné que la majorité des données sur le web ne sont pas structurées. Une approche prometteuse pour avoir un accès programmable à ces connaissances est l'utilisation de techniques d'extraction d'information. Le plus souvent, ces techniques visent à extraire les entités, comme les personnes, lieux géographiques, etc., partir de texte libre. Ces entités peuvent potentiellement être reliées les unes aux autres, créant ainsi de facto un graphe de connaissance globale des entités liées. Un certain nombre de défis liés aux entités doivent être abordés pour la réalisation d'un point de vue orienté vers les entités du Web sémantique. Dans cette thèse de doctorat, nous contribuons à la recherche dans le domaine de l'extraction d'entités à partir de documents de texte en ligne avec pour objectif de faciliter l'adoption du Web sémantique. Cette thèse aborde les problèmes liés aux entités suivants: *Résolution d'entité pour les documents Web*, *Reconnaissance d'entité dans les environnements micro-blogging*, et *Profilage d'entité et Applications*. Plus précisément, nous faisons les contributions suivantes:

(1) ***Résolution d'entité pour les documents Web:*** L'un des principaux défis pour réaliser un traitement automatisé de l'information sur le Web est lié au problème de la résolution d'entité. Il y a un certain nombre d'outils qui parviennent à reconnaitre de manière fiable des entités nommées, comme les personnes, les entreprises, ou les lieux géographiques, dans les documents Web. Les noms de ces entités extraites ne sont cependant pas uniques; le même nom sur différentes pages Web peut ou ne peut pas faire référence la même entité. Nous abordons ce problème d'homonymie, qui est très semblable au problème de la résolution d'entité, étudié dans les bases de données relationnelles, mais comporte aussi quelques différences. Mais surtout les pages Web ne contiennent souvent que des informations partielles ou incomplètes sur les entités. Nous proposons donc un cadre général où plusieurs fonctions de similarité correspondant aux règles spécifiques du domaine peuvent être définies. Nous utilisons des techniques de la théorie des graphes et d'apprentissage pour combiner efficacement les

données provenant de multiples fonctions de similarité afin d'améliorer la reconnaissance d'entité, et de démontrer l'efficacité de notre méthode sur deux ensembles de données du monde réel.

(2) ***Entités sur les micro-blogs comme Twitter:*** Twitter est un service de micro-blogging populaire sur le Web, où les gens peuvent entrer des messages courts, qui deviennent alors visibles par les autres utilisateurs du service. Bien que les sujets de ces messages varient, il y a beaucoup de messages où les utilisateurs expriment leurs opinions sur les sociétés ou leurs produits. Ces messages sont une riche source d'information pour les entreprises pour l'analyse des sentiments ou des opinions. Il y a cependant un grand obstacle avant d'analyser directement les messages: comme les noms de société sont souvent ambigus (par exemple, la pomme (apple), le fruit vs Apple Inc.), il faut d'abord identifier les messages qui sont liés à la société en question. Nous présentons tout d'abord des techniques simples qui font usage de profils d'entreprise, que nous avons créés semi automatiquement à partir de sources Web externes. Nos techniques avancées prennent en compte une estimation de l'ambiguïté et complètent automatiquement les profils d'entreprise grâce à un apprentissage actif sur le flux Twitter lui-même. Nous démontrons l'efficacité de nos méthodes à travers un vaste ensemble d'expériences. Nous présentons aussi TweetSpector comme un prototype fonctionnel de classification axée sur les entités de tweets.

(3) ***Profilage d'entité et Applications:*** Le profilage d'entité est le problème de la construction d'une représentation compacte (profil) d'une entité, qui résume les différentes mentions d'une entité. Nous nous concentrons sur la construction d'un profil d'une entité d'un utilisateur et une entité de localisation, et montrons des applications qui font usage de ces profils d'entité.

   (a) ***Profils d'entité d'utilisateurs sur les réseaux sociaux:*** Les utilisateurs, par le biais de leurs activités sur les réseaux sociaux, laissent des traces de leur personnalité. Avec les progrès de l'extraction de contenu et des techniques de modélisation, il devrait être possible de profiler l'entité de l'utilisateur à partir de son contenu sur un réseau social. Dans ce travail, nous explorons différentes techniques pour résumer une présence de l'utilisateur sur les différents réseaux sociaux. Nous montrons que l'un des avantages de maintenir un profil de l'utilisateur est de fournir le contexte pour comprendre les textes courts, et d'aider une meilleure compréhension des micro-messages. En outre, nous présentons TripEneer: un outil de recommandation de planification de voyage basé sur le profil de l'entité de l'utilisateur et de l'entité de localisation.

   (b) ***Fusion de données sociales et issues de capteurs d'une entité de localisation dans le Cloud:*** Comme l'informatique mobile en nuage facilite un large éventail d'applications intelligentes, la nécessité d'une fusion des divers types de données disponibles dans le nuage se développe rapidement. En particulier, les données sociales et issues de capteurs sont au cœur de ces applications, mais sont généralement traitées séparément. Ce travail explore le potentiel de fusion des données sociales et issues de capteurs, liés à une entité de localisation, dans le nuage. Nous

présentons un système de recommandation de voyage qui repose sur un cadre conceptuel. Ce cadre permet de mélanger les données sociales et issues de capteurs hétérogènes pour une analyse intégrée, comme par exemple l'extraction de l'humeur des gens en fonction des conditions météorologiques à partir de Twitter et des flux de données de capteurs météorologiques.

Cette thèse, grâce à ses contributions permettant de relier les entités sur le web, fait une étape prometteuse vers la réalisation d'une vue du Web (sémantique) axée sur les entités.

Mots-clés: *entité, résolution d'entité, profil d'entités, recommandations, web sémantique, flux twitter, réseaux sociaux, apprentissage automatique, gestion de la réputation en ligne, mesure de sentiment, modèles d'utilisateur*

Contents

List of Figures

List of Tables

List of Algorithms

Part I

Introduction

Chapter 1

Introduction

I have a dream for the Web in which computers become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A "Semantic Web", which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The "intelligent agents" people have touted for ages will finally materialize.

Tim Berners-Lee vision of the Semantic Web

1.1 Background

Invention of Internet and World Wide Web (WWW) revolutionized the way information is produced and consumed across the globe. The ease of creation of web documents and interconnecting these web documents using hyperlinks is the primary reason that ignited the massive growth that resulted in the Web that we know today. Algorithms like PageRank [PBMW99] and HITS [Kle99], which exploit these simple interconnections (hyperlinks) across web documents, are enabling keyword-based search engines to readily identify subset of documents that are relevant to the user query. However, most of the human queries still require the human to refer to multiple web documents resulting from the search engine and use his intelligence to connect the information across the documents to meet his needs. As the needed information is already present in the web documents as free text, it could be a great advantage for the computers to extract this information, possibly understand the semantics, interlink the information across documents, do the inference and provide the user with the needed information. The primary requirement to achieve the goal of computers automatically connecting information across the web documents is that these documents should be augmented with meta-data, followed by developing systems which can use this meta-data to connect information across web documents. The core vision of Semantic Web is to extend this simple links among documents to interlinks of data/information across documents, and providing a programmatic access to such knowledge.

Semantic Web is a broad concept and could refer to multiple things. In some scenarios it is referred to as web of data, where the information related to a document is represented using RDF [RDF], microformats (hCard, hCalendar, XCN, etc.) [MF], etc. Many times it is also referred to the technology stack that enables computers to automatically interlink information across data sources. Semantic Web also refers to the web services offering semantic functionalities. Semantic Web can also be seen as a global database of web documents against which one can run queries using SPARQL [SPA], and apply inference and reasoning using OWL, Ontologies etc. All in all, Semantic Web encompasses all the technologies that can extract semantics (knowledge) from data. Research in semantic technologies involves developing the various components in the stack: developing meta-data annotation tools, enabling querying and inferring on these meta-data stores (SPARQL, RDF-DB Engines), Vocabularies, Ontologies (OWL [OWL]) and Schemas for AI inference.

The efforts of W3C (World Wide Web Consortium) on Semantic Web resulted in a number of standards for representing the meta-data. On the one hand there is the comprehensive, heavy-weight, and graph-based RDF, while on the other hand there are simplistic micro-formats (hCard, hCalendar, XCN, etc.). For the core of semantic techniques to work it is essential that the web documents should be annotated with meta-data. Given the huge number of web documents, there are two natural approaches for adding meta-data to the documents.

- *Manual Approach*: This first approach involves data providers manually enriching their webpages with metadata in one of the standard metadata formats (RDF or micro-formats), so that it is machine readable. Some big data providers (like Government organizations, Yahoo, etc.) are making their data available in this format. The incentive for the data providers to annotate their content is that their data can be consumed by a wider audience. Tools like Dapper [Dap], Semantify [Sem], etc., are aiding publishers to add semantic annotations to their existing web documents. For a technology to be massively adopted by the users, it is important that the technology is simple and easy to integrate into the existing infrastructure. Despite the existence of tools and incentives, it is difficult to make this approach widespread because of the involved manual effort. This is a very time-consuming process, and to adopt it for web-scale is a daunting task. Examples of data-sources based on this approach include: DBpedia [DBp] (semantic version of Wikipedia), FreeBase [Fre], many data sources released by Governments [Gov], and Nepomuk [Nep] (a social semantic desktop).

- *Automatic Approach*: The success of Google is that it understands the webpages as is, without the demand that the web documents publishers follow all the W3C recommendations. For the Semantic Web to be popular it is important that any proposed approach understands the web documents as they are. In view of this, the second approach proposes to infer the metadata automatically from the existing webpages. Additionally, semantic tools based on these automatic approaches are equipped on dealing with imperfections and uncertainties in existing information. Examples of contributions based on this approach include: the natural language based processing tools that do entity extraction - such as Calais [Ope] and TextWise [Tex] APIs that recognize entities like people, companies, places, etc. in documents; vertical search engines, like ZoomInfo [Inf], 123People

[lin] and Zaba [Sea], which mine the web for people; technologies like GATE [GAT] and Apache UIMA [UIM], which recognize objects in web pages. These two approaches, *manual* and *automatic*, can compliment each other. The partial metadata added using manual approach can be used by automatic approach to extract other metadata from a document.

The main question in the *second approach* concerns the kind of information that can be automatically extracted from the web documents. It is very unlikely to automatically extract every possible information from a document accurately. It could be argued that most of the web documents (news articles, blog entries, wiki articles, etc.) are about entities, their descriptions, relationships among entities, attributes of entities, concepts, and relationships between concepts. Automatic extraction of information related to entities has been the central theme of many conferences like MUC (Message Understanding Conferences) [GS96] and TREC [TRE] over the years. This thesis focuses on several entity related problems, solving which makes important steps towards realizing the entity-oriented view of the Semantic Web [BSG07].

The concept of entities is not new, and is quite central to many computer systems and database management systems. The tables in popular relational databases are usually modeled on concept of entities. In software engineering *Entity Relationship* (*ER*) model is used as an abstract way of describing the database. This popular Entity Relationships design methodology, based on entities, aids in building efficient computer systems.

The data available on the web is heterogeneous and is usually structured, semi-structured or unstructured data. However, according to International Data Corporation (IDC) 80% of the data on the web is unstructured and is growing at rapid pace due to the ease with which data can be published on blogs, media exchanged on social networks, etc. Semantic Web can be seen as an effort to convert all the unstructured data on the web to structured format that could be queried upon. Once converted to structured format, the data can be efficiently handled using successful DBMS technologies. In this thesis work we focus on entity related problems in unstructured documents.

1.2 Entities on the Web and the Challenges

What is an entity? As per Wikipedia an entity is something which exists by itself, although it need not be of material existence. Some examples of entity include: any real or fictional person, location, company, organization, product, object, etc. An entity is usually described with set of attributes. For example: a person is described using name, date of birth, gender, etc.; a location is described by name, co-ordinates, etc.; a book by its title, author, publisher, etc. Majority of the web documents – like homepages, news articles, blog entries, websites, comments, microposts, etc. – are about entities.

Entity-oriented view of Semantic Web involves linking entities on the Web. Such a web would empower users with easy information discovery and provide alternate ways of exploring the Semantic Web. A number of entity related challenges need to be addressed for realizing such an Entity-oriented view of Semantic Web (Figure 1.1).

Entity Extraction: Human language is rich, complex and ambiguous. The web documents containing the information about entities are usually expressed in the natural language format. This poses a number of challenges for computers to automatically extract information related to entities. The

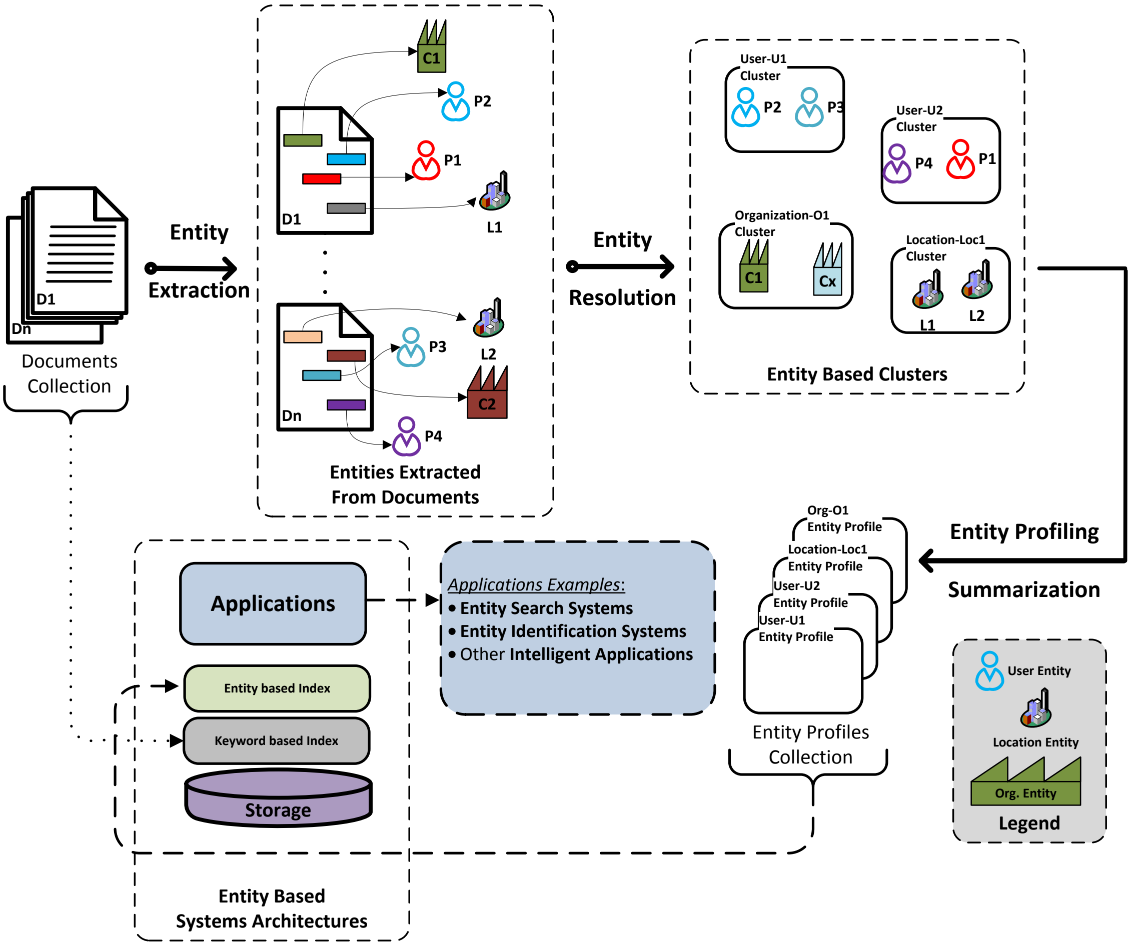


Figure 1.1: Entity Oriented View of Semantic Web

first entity-related challenge is *entity extraction*. Entity extraction is about identification of entities that are being described in a free text. It is commonly used to parse unstructured text documents and extract useful entity information (like location, person, brand, etc.) to construct a more useful structured representation of the entity. Entity extraction is one of the most common text preprocessing tasks to automatically understand a text document. If the entity extraction is concerned about only certain type of entities then it is often referred to as *Named Entity Extraction* (*NER*) [NER] in literature. The NER tools developed for text documents work on one document at a time for an entity extraction. The NER tools for web documents face additional challenges of dealing with multiple web documents at a time and the uncertainties involved in text extraction from the html content. There is also need for developing NER tools for micro-blogging kind of media, as the language on these media is very dynamic.

Entity Resolution: The next challenge related to entities is *Entity Resolution*. A real world entity (e.g. a person, book, etc.) is described on the web at multiple pages possibly in different ways. For example, a person entity *Tim Berners Lee*, inventor of the world wide web, could be described as Tim Berner on one webpage and as Tim B. Lee on another web page. *Entity Resolution* (*ER*) is the problem of deciding whether two or more entity mentions (Tim Berner, Tim B. Lee) refer to the same real world entity (Tim Berners Lee). This is a common problem in many information integration scenarios, where multiple data sources might contain different representations of the same entity. The attributes of an entity description at one source (for example on a webpage) can be different from describing the same entity at a different source (another webpage). Another example of this situation is an eProfile management system, where a user (customer) can have different profiles at different websites. Also, the amount of information about an entity can be different at the different sources. Entity Resolution addresses the problem of linking these different profiles to the same user entity.

More examples of entity resolution applications include: large customer-oriented organizations often need to merge long lists of names and addresses (possibly obtained from different data sources) in order to get to know more about its customers; web portals, like CiteSeer [Cit], Cora [Cor], etc., they need to integrate citations and paper titles, parsed and extracted from several personal and publisher web-pages.

Entity Resolution leads to efficient organization of data/knowledge, and information enhancement (knowledge discovery) of an entity when the data from the diverse sources is reconciled. However entity resolution is a difficult but an important problem. Some of the underlying challenges are due to the lack of an explicit structure for representing entities. Each source has its own way of representing an entity. There is no unique structure or schema for the entities. The other challenge is the uncertainties involved in the entity representation. Uncertainties may arise due to typographical errors, non-standard representation, missing information, inherent ambiguity, malicious intent, the trustfulness of the source, etc. Given that Entity Resolution problem is a challenging task, there is scope for better theoretical models, better algorithms or techniques which are computationally feasible and achieve more accuracy. There is need for research in these directions in order to realize efficient systems.

Entity Identification: The linking of entities across data sources will be easier if each entity maintains a unique entity-id. When RDF format is used for representing metadata, it is common to use URI as an unique id for a resource (aka entity). For a small repository, it should be possible to have unique ids for the entities. How does one ensure the entity ids are unique for an entity across the web? How does one build such an Entity Identification System (EIS)?

OKKAM[1] [BSG07] is an example of such an EIS system. OKKAM envisions an Entity Name System (ENS) that can act as a repository of the entities on the web, where each entity would have an unique entity id. OKKAM's guiding philosophy is "*Entity identifiers should not be multiplied beyond necessity.*" Newly generated content on the web can use existing entity identifiers if the described entities are already present in the ENS; if not, a new entity identifier will be generated by the ENS, which can be used from there on. Thus OKKAM would act as a global service for providing the unique identifiers for the entities on the web.Such a service would power entity-centric search engines, enable easier and efficient web-scale mash-ups, improve the quality of the content production in professional environments, etc.

What kind of entity models are ideal for such an Entity Identification system? How can the EIS handle the life-time evolution of an entity, where new information related to the entity gets added and some past information is valid no-more or conflicting with the existing information? EIS

[1] http://www.okkam.org

would need ER algorithms for maintaining unique entities in its repository. Algorithms can rely on such entity identification systems to develop more efficient ER techniques.

Entity Summarization and Profiling: Once we identify entity mentions as related to the same entity using the ER techniques, we face the problem of consolidating the various information related to entities. Should we simply merge all the information and consider it as one entity, or maintain the history of the evolution of the entity. For efficient storage solutions it might be ideal to merge all the information and discard all the other information. During consolidation how does one handle conflicting information? If one is interested in the provenance of the entity data, then the system should maintain all the information.

What kind of entity model is efficient for a summarized view of the entity? It could be a simple bag-of-words model, semi-structured attribute-value pairs, or extensive graph-based model (RDF [RDF]). How does one go about consolidating conflicting information across different sources. A user entity is present on the web across many websites. If a system can maintain a compact summary of the user entity, many customized services related to the entity can be constructed. All the queries can use the user entity profile as a context for the queries, thus providing user entity centric results.

Entity Search Systems and Entity-based Retrieval: Entity-based information retrieval received a lot of attention at major conferences like TREC [TRE], INEX [INE], etc., over the past decade. The challenges here include: how does one index or organize the document collection in order to retrieve entities specific to the user query; what are the query models for entity-specific search; and what are efficient metrics for evaluating entity search systems?

Twitter, Facebook, etc., are important platforms where users publish their opinions, emotions, and news about products, tv-shows, sports, books, companies, etc., they interact with. This information is of tremendous value for companies, organizations, etc, for studying the trends, for making informed decisions. From this deluge of data it is essential to retrieve data relevant to a specific individual or organization entity. How does such an entity-based retrieval system be designed that can work at large scale and in real-time? Current search systems interface are keyword-based. Efficient query interfaces would be needed to support entity-based search.

Uncertainties and Philosophical Ambiguities: There is also a philosophical aspect to Entity and its existence. It is not possible to give a precise definition for an entity. In the popular super-hero series Superman, the main character has two different personalities: Clark Kent (alter-ego) and Superman. Do these two personalities refer to the same entity or are these two different entities? A person during childhood and the same person in his old age, each with different personality, are they two different entities or the same person entity? A sub-part of an entity, should it be treated as an independent entity, or should it be treated as an attribute of the main entity? There is also a philosophical debate about the existence of abstract entities and their differences with concrete entities. We do not dwell into these involved philosophical aspects of the entities in this thesis.

The success of research efforts in the fields of Machine Learning, Natural Language Technologies, Artificial Intelligence, and Database technologies have to come together in order to address these challenges.

1.3 Contributions

Having presented the main challenges involved in realizing Web of Entities, we now give an overview of the challenges addressed in this work. The thesis broadly focuses on the following problems: Entity Resolution, Entity Matching and Entity Profiling.

1.3.1 Entity Resolution for Web Documents

One of the key challenges to realize automated processing of the information on the Web, which is the central goal of the Semantic Web, is related to the entity resolution problem. There are a number of tools that reliably recognize named entities – such as persons, companies, geographic locations – in Web documents. The names of these extracted entities are however non-unique; the same name on different Web pages might or might not refer to the same entity. The entity resolution problem concerns of identifying the entities, which are referring to the same real-world entity. This problem is very similar to the entity resolution problem studied in relational databases. However, there are also several differences. Most importantly Web pages, often only contain partial or incomplete information about the entities.

Similarity functions try to capture the degree of belief about the equivalence of two entities, thus they play a crucial role in entity matching. The accuracy of the similarity functions highly depends on the applied assessment techniques, but also on some specific features of the entities. We propose systematic design strategies for combined similarity functions in this context. Our method relies on the combination of multiple pieces of evidence, with the help of estimated quality of the individual similarity values and with particular attention to missing information that is common in Web context. We study the effectiveness of our method in two specific instances of the general entity matching problem, namely the person name disambiguation and the Twitter message classification problem. In both cases, using our techniques in a very simple algorithmic framework, we obtained better results than the state-of-the-art methods. Specifically, we make following contributions:

- We layout the similarities and differences that exists between the ER problem for structured database records and the ER problem for unstructured documents
- As the ER algorithms depend on domain specific rules, we propose a generic framework where in multiple similarity functions corresponding to the domain specific rules can be defined. We make use of techniques from graph theory and machine learning for efficiently combining the evidence from multiple similarity functions for improved ER results.
- We demonstrate the efficiency of our framework by applying our techniques on two real-world datasets. (i) Web people names dataset and (ii) Twitter dataset.

1.3.2 Entity Matching on Twitter

Twitter is a popular micro-blogging service on the Web, where people can enter short messages, which then become visible to other users of the service. While the topics of these messages varies, there are a lot of messages where the users express their opinions about some companies or their products. These messages are a rich source of information for companies for sentiment analysis or opinion mining. There is however a great obstacle for analyzing the messages directly: as the company names are often ambiguous (e.g. apple, the fruit vs. Apple Inc.), one needs first to identify, which messages are related to the company. In this part we address this question. We present various techniques for classifying tweet messages containing a given keyword, whether they are related to a particular company with that name or not. We first present simple techniques, which make use of company profiles, which we created semi-automatically from external Web sources. Our advanced techniques take ambiguity estimations into account and also automatically extend the company profiles from the twitter stream itself. We demonstrate the effectiveness of our methods through an extensive set of experiments. Moreover, we extensively analyze the sources of errors in the classification. The analysis not only brings further improvement, but also enables to use the human input more efficiently. Our contributions include:

- We address the challenge of matching a named-entity in a short message (twitter message) by focusing on semi-automatic creation of entity profile.
- Entity profile of a named-entity is constructed using data mined from multiple data sources. This entity profile already outperforms the state-of-art methods created for classifying tweet messages containing the entity mention.
- We make use of estimated *relatedness-factor* to further improve the classification performance of entity profile based classification.
- We resort to Active Learning for continuously updating the entity profile by monitoring the live Twitter streams.
- We present *Tweetspector* as a working prototype for entity-based classification of tweets in real-time.

1.3.3 Entity Profiling and Applications

We have seen that an entity (a user, a location, etc.) is mentioned on multiple web documents (web pages, twitter streams, news, etc.). Entity profiling is the problem of constructing a compact representation (profile) of an entity, which summarizes the various mentions of the entity. Firstly, we focus on constructing an entity profile of an user entity, and show applications that make use of such constructed user entity profiles. Next, we look at a location entity profile based travel recommendation system, where we fuse social and sensor data corresponding to a location.

- Pervasive web and social networks are becoming part of everyone's life. Users through their activities on these networks are leaving traces of their personalities. With the advances of content mining and modeling techniques it should be possible to profile the user entity. In this work we explore various techniques for summarizing a users online presence. In this work we show that one of the advantages of maintaining a user profile is to provide a context for understanding the short texts, and help in better understanding of microposts. Our contributions are:
  - We present a number of state-of-art techniques in order to extract the main topics of the content published by a user on his online social networks.
  - The merits and demerits of each of the entity profile creation are presented.
  - Dataset specific to evaluate the entity profile is created and evaluated.

  - Using the user-specific profile we present the enhanced named-entity recognition in Twitter context.
  - In *TripEneer* (User social profile based travel plan recommendation) :
    * We profile both the user-entity and location-entity (travel destinations) and present various ranking schemes which allow the user to choose his travel destinations easily and efficiently.
    * We present a scalable user-based travel plan recommendation system.
    * Our application enables the user to choose his places of interests easily and efficiently compared to the current systems.

- As mobile cloud computing facilitates a wide spectrum of smart applications, the need for fusing various types of data available in the cloud grows rapidly. In particular, social and sensor data lie at the core in such applications, but typically processed separately. This work explores the potential of fusing social and sensor data of a location entity in the cloud, presenting a practice—a travel recommendation system that offers the predicted mood information of people on where and when users wish to travel. The system is built upon a conceptual framework that allows to blend the heterogeneous social and sensor data for integrated analysis, extracting weather-dependent people's mood information from Twitter and meteorological sensor data streams. In order to handle massively streaming data, the system employs various cloud-serving systems, such as Hadoop, HBase, and GSN. Using this scalable system, we performed heavy ETL as well as filtering jobs, resulting in 12 million tweets over four months. We then derived a rich set of interesting findings through the data fusion, proving that our approach is effective and scalable, which can serve as an important basis in fusing social and sensor data in the cloud. We make following contributions:

  - We present techniques to extract the social metrics of a location based on the real-time tweets.
  - A scalable system is presented for fusing the social metrics with sensor metrics of a location, and providing the user with enhanced contextual information.

1.4 Thesis Organization

The remainder of the thesis is organized as follows: Chapter 2 presents a survey of literature related to research challenges addressed in this thesis work. In Chapter 3, we address the problem of Entity Resolution of the mentioned person-names in web-documents. We present a generic framework in which (a) abstract similarity functions that capture the degree of belief about the equivalence of two entities mentioned in unstructured documents, can be defined; (b) results from various similarity functions are combined; (c) efficient ER techniques can be designed by using the results from graph modeling and machine learning. Accurately constructing the context around the entity mention directly impacts the performance. Next, in Chapter 4, we address the problem of retrieving short messages/microposts containing entity mention that actually matches with a specific entity . Microposts being short messages contain little or no-context, thus making it a challenging problem. The proposed solutions focus on rich construction of the entity profile. We also demonstrate *Tweetspector* (in Section 4.8): a prototype for entity-based classification of tweets. Chapter 5 is about user entity profiling, where we discuss several techniques for accurately constructing a user profile based on the users presence on a number of social networks. We additionally demonstrate *TripEneer*: Travel recommendation system based on user entity profile. Chapter 6 presents social and sensor data fusion of a location-entity in the cloud. Finally we summarize the conclusions and discuss the future work in Chapter 7.

1.5 Selected Publications

This thesis is based on the following several research papers that were published during the course of this work.

- S. R. Yerva, Z. Miklós, and K. Aberer, “Towards better entity resolution techniques for Web document collections,” in *1st International Workshop on Data Engineering meets the Semantic Web (DESWeb'2010) (co-located with ICDE'2010)*, Long Beach, California, 2010.
- S. R. Yerva, Z. Miklos, and K. Aberer, “It was easy, when apples and blackberries were only fruits,” in *Third WePS Evaluation Workshop: Searching Information about Entities in the Web, CLEF (Notebook Papers/LABs/Workshops)*, Padova, Italy, 2010.
- S. R. Yerva, Z. Miklos, and K. Aberer, “What have fruits to do with technology? The case of Orange, Blackberry and Apple,” in *International Conference on Web Intelligence, Mining and Semantics (WIMS 2011)*, p. 48, ACM, Sogndal, Norway, 2011.
- S. R. Yerva, Z. Miklos, and K. Aberer, “Entity-based classification of twitter messages,” *International Journal of Computer Science and Applications*, 2012.
- S. R. Yerva, H. Y. Jeung, and K. Aberer, “Cloud based Social and Sensor Data Fusion,” in *FUSION*, Singapore, 2012.
- S. R. Yerva, Z. Miklos, and K. Aberer, “Quality-aware similarity assessment for entity matching in Web data,” *Semantic Web Data Management, Elsevier Information Systems Journal*, 2012.
- S. R. Yerva, J. Saltarin, H. Y. Jeung, and K. Aberer, “Social and Sensor Data Fusion in the Cloud,” in *Mobile Data Management*, Bangalore, India, 2012.
- S. R. Yerva, Z. Miklos, F. A. Grosan, O. A. Tandrau, and K. Aberer, “TweetSpector: Entity-based retrieval of Tweets,” in *SIGIR*, Portland, USA, 2012.
- S. R. Yerva, F. A. Grosan, A. O. Tandrau, and K. Aberer, “TripEneer: User-based Travel Plan Recommendation Application,” in *7th International AAAI Conference on Weblogs and Social Media*, Boston, USA, 2013.
- S. R. Yerva, M. Catasta, G. Demartini, and K. Aberer, “Entity disambiguation in Tweets leveraging User Social Profiles,” in *IEEE 13th International Conference on Information Reuse & Integration, IRI 2013*, San Francisco, CA, USA, August 14-16, 2013.

Chapter 2

State of the Art

If I have seen farther, it is by standing on the shoulders of giants.

Issac Newton

2.1 Introduction

In this chapter, we present the state-of-the-art work related to the thesis in detail. We mainly review the works related to the entity related problems: *Entity Resolution*, *Entities in Twitter streams*, and *User Entity Profiling*. Entity Resolution is an important problem and it appears in many scenarios, thus, has received significant attention over last 20-30 years. First, we present how Entity Resolution problem was addressed in databases and citations domains, next we also look into number of differences among these approaches, and finally how it is addressed for web document collections in a scalable and distributed manner.

With the popularity of social networks, micro-blogging platforms and e-commerce, more and more relevant data is becoming accessible on the Web. We review various research efforts which mined these real-time data – with special emphasis on twitter data containing entity mentions. After covering the works related to generic entities on the Web and Twitter streams, we shift our focus on modeling the most important entity i.e. the user entity. We present a number of works which have modeled the user and number of applications that have benefited from the user models. Further we present works related to information fusion based on the location entity.

2.2 Entity Resolution

Entity Resolution (ER) is the problem of identifying and merging entity mentions, appearing in database records or documents, referring to the same real-world entity. It has received significant attention in the literature over last 20-30 years. It is an important problem in the domains that involve data cleansing [Mj03] and information integration [HS09]. Some ER motivating examples include: linking census records, public health records, comparison shopping, counter terrorism, spam detection, web mashups, linking paper citations, etc.

Entity Resolution is the problem of identifying and merging entity mentions, appearing in database records or documents, referring to the same real-world entity. This is a challenging task owing to a number of reasons.

1. *Variations in text representations:* A single entity, for example a person entity: Michael Jordan, could be named in various ways: M. Jordan, Michael J., Michele Jordan, etc. There could be number of reasons for such variations: convention used by the content publisher, typographical errors, impromptu acronyms, colloquial usage, etc. Since it is essential that ER algorithms should handle such text variations, it is hardly a surprise that the string similarity functions form a core component of the ER techniques. String similarity functions can be broadly grouped into: *Edit Distance based*: Jaro [Jar89, Jar95], Wrinkler [Win99], Monge & Elkan [ME96, ME97]; *Set similarity based*: Jaccard, Dice, etc.;*Vector based*: cosine similarity, tfidf, etc.; *miscellaneous*: phonetic based soundex [Zob96], fuzzy matching similarity [CGM05].

2. *Non standardization:* Since there is no standard agreed-on notation for representing the information, various conventions are used by various publishers. Usage of acronyms, short forms, etc., are some of the reasons which make Entity Resolution a tough problem. A person name could be represented as first name, or last name, or some combination of first and last name; address of a location could be represented in multiple ways; attribute names could be used differently at different places; the same date could be represented in multiple formats. The ER techniques rely on converting an entity mention to certain standard format, and make comparisons in this standard format. ER algorithms could rely on schema alignment and schema matching techniques.

3. *Missing Values and Missing Context:* Many of the real world entities could have same entity mention names, which is also one of the reasons for ER being a challenging task. A person name like Michael Jordan could refer to the popular basket ball player[1], or the famous Machine Learning researcher[2], or some other person entity. So, it is difficult to identify who the real entity is until more information is available. It is essential that an entity mention will be resolvable only when the context around entity mention is clear. Many ER techniques have been proposed which vary in how they handle the context during the disambiguation process.

4. *Uncertainties and Constraints:* The ER algorithms should be robust enough to handle inherent uncertainties in the entire process. Uncertainties could arise due to automatic error-prone techniques, errors in data preparation steps, threshold deciding steps, missing attributes, etc. A number of simple constraints may fail. For example, when record M_1 matches with M_2, record M_2 matches with M_3, but M_1 does not match with M_3 according to an ER technique. ER techniques should be clever enough to handle such conflicts by enforcing the constraints.

5. *Big Data:* Given the deluge of data, there is a need for efficient ER techniques that can exploit parallelism and use pruning techniques to keep lower computational costs . Given the recent advances in handling big-data via the map-reduce technologies, there is need for adapting ER techniques than can make use of map-reduce frameworks. With larger data comes additional challenges of: dealing with heterogeneous data (unstructured, semi-structured, unclean, and incomplete data);

[1] Basket ball player: `http://en.wikipedia.org/wiki/Michael_Jordan`

[2] Machine Learning and AI Researcher: `http://en.wikipedia.org/wiki/Michael_I._Jordan`

additionally infer links and relationships besides equality among entities; and additionally deal with multi-domain data.

AI community has proposed different models for representing knowledge. [TY94] provides the information processing with a theoretical foundation. The four basic notions of this model are *data*, *information*, *knowledge* and *wisdom*. The model defines transformation from one notion to another as similar to humans gaining information from data, knowledge from information, and wisdom from knowledge. The current computer systems mimic these transformations. It refers to this transformation as entropy reduction process. The process of entity resolution can be seen as an important step in entropy reduction process.

ER Variations and Techniques

ER is the problem of identifying and grouping different manifestations of the same real world object. The Entity Resolution (ER) problem or similar variations have been referred to in literature as record linkage [Win99], merge/purge [HS95], reference reconciliation [DHM05], de-duplication [SB02], reference matching [MNU00], object identification [TKM02], co-reference resolution [SNL01] and identity uncertainty [MW03].

Record Linkage [KSS06] is the problem of linking records that refer to the same entity across data sources (e.g., data files, books, citations, websites, databases). It is a challenging task as these different records may not share a common identifier like URI, database key, or unique id, etc. Reference reconciliation [DHM05] is the problem of identifying when different references (i.e., sets of attribute values) in a dataset correspond to the same real-world entity. De-duplication [SB02] corresponds to clustering the records or documents, containing entity mentions, based on the same entity. Each such cluster would be representative of that real world entity. Reference matching [MNU00], another variation of ER, refers to the problem of linking noisy records to clean records in a reference table.

The research efforts in the field of entity resolution have been at different levels. Research efforts were proposed to formalize the theory by providing mathematical models [FS69, BGMM^{+}09]. Newcombe et al. [NKAJ59] are among the first who introduced odds ratios of frequencies and decision rules for delineating matches and non-matches. Fellegi and Sunter [FS69] provided mathematical models that define the odds ratios of frequencies as R. Given the thresholds, they propose a decision rule which decides if two records are matching, non-matching or are uncertain, based on the relation of R with respect to the defined thresholds. The models proposed in these papers form a central part of many modern ER techniques proposed in literature.

On other side, efforts were put in to solve the entity resolution problem by coming-up with techniques that are computationally efficient, accurate, results and rules interpretable by humans, and scalable. Earlier solutions to the entity resolution problem were based on the closeness/similarity of the attributes of the entities. As majority of the attributes are of string format, many efficient string matching algorithms [Coh01] [CRF03] were proposed. Further ER solutions were based on manually set rules as in [HS95]. Subsequent work focused on learning the rules [TKM02] [SB02]. Some works used clustering as a forte in their ER algorithms. All the ER techniques can be seen as *pairwise ER* (comparing records in pairs) or *collective ER* (the whole dataset is resolved by the use of clustering techniques). Techniques and results from machine learning are applied extensively to the entity resolution problem.

Matching and merging functions [BGMM$^+$09] are two important abstract functions of the many generic ER techniques. Clever matching and merging techniques have been proposed. Again, the focus was making these functions easily applicable and computationally efficient on the dataset at hand. In the paper [BGMM$^+$09] these functions are considered as black boxes. The calls to these black boxes are usually expensive. So, the focus of the proposed swoosh algorithms was to make calls to these functions as less as possible. Authors identify four core fundamental properties: ICAR (*Idempotence, Commutative, Associative, and Representativity*). Efficient algorithms were proposed depending on the properties that "*match and merge*" functions satisfy. Some research efforts have focused on incorporating constraints in their ER techniques to avoid inconsistencies and identify participating source trustfulness.

In the following sections we look, in depth, into these different efforts. We discuss *pairwise vs collective ER* techniques in Section 2.2.1, *constraints-based ER* in Section 2.2.2, *big-data and distributed ER* in Section 2.2.3 and finally in Section 2.2.4 we present techniques specific to *ER for web documents* .

2.2.1 Pairwise vs. Collective ER

The ER techniques proposed in literature can be broadly classified into two categories: *pairwise ER* and *collective ER*, depending on whether the pair of records are compared and decided if they refer to the same entity or if the decision is made on collection of records. Pairwise ER techniques typically involve comparing two records, and based on the component-wise similarities of the comparison vector decide if the two records match or do-not-match. These techniques could rely on hand coded rules, manually set thresholds, or the thresholds decided by the use of machine learning approaches. Fellegi and Sunter [FS69] were one of the first to formalize this approach, which has formed the basis for many works that followed.

Machine Learning (ML) approaches were extensively used in pairwise matching ER algorithms mainly to decide the weights for individual components of the comparison vector and in deciding the suitable thresholds for separating matches from non-matches. Machine Learning techniques like Decision Trees [CKLS01], SVMs [BM03, Chr08], ensemble of classifiers [CKM09], conditional random fields [GS09], etc., have been used for solving pairwise ER problem. Even though the ML based techniques are more effective than hand coded rules, their efficiency is dependent on the availability of the training sets which usually involves a significant cost to obtain. Further, one has to deal with class imbalance problem. It is quite common to have many more negative examples (non-matches) compared to positive examples (matches).

A number of techniques have been proposed either to avoid the supervised training sets, or to minimize the number of trained examples, or to employ crowd sourcing for creating training sets. Alternatives to the above supervised learning ER techniques include unsupervised or semi-supervised techniques like: Expectation Maximization (EM) based techniques to learn parameters [HSW07, WWP06] and Generative Models [RC04].

Active learning is a supervised machine learning technique where the learning algorithm is able to interactively query the user (or other information source) to obtain the desired output for selected examples. It is ideal for situations in which unlabeled data is abundant but manually labeling it is expensive. Since the learner chooses the examples, the number of examples to learn a concept can often be much lower than the number required in normal supervised learning. Active learning is used in ER works like: learning from a committee of classifiers [SB02, TKM02], ER techniques based on optimizing precision and recall [AGK10, BIPR12, BHLZ10], using crowd sourcing for labeling selective examples [WKFF12, MWK$^+$11].

Deduplication [SB02] similar to ER, aims at detecting and eliminating duplicate records that refer to the same entity. Traditional systems perform deduplication by use of hand-coded functions or rules coded by the domain expert. This task is challenging and non-trivial. The paper argues the need for automation/machine learning techniques. Machine learning techniques involve designing a classifier with the use of training data. The quality of the training set decides the effectiveness of the classifier. This paper proposes efficient Active Learning technique, which would identify decision rules for deduplication.

Collective ER techniques make resolution decisions on a group of records rather than each pair of records. These algorithms often employ variety of clustering algorithms for grouping records, which may take pair-wise similarity graph as input. Some of the frequently used cluster methods for ER include: hierarchical clustering [BBS05], nearest neighbor based clustering [CGM05], correlation clustering [SNL01, BBC04, NC02, EC08, ES09, ACN08], etc. It may also require construction of a cluster representative or canonical entity representing the cluster with maximal information [BGMM$^+$09, DBES09, CWH$^+$07, PRMB12].

Collective ER decides on a collection of records by placing similar records, that could potentially refer to the same entity, in the same cluster. These techniques progress either by joining clusters or breaking existing clusters depending on the seen evidence. Decisions for one cluster membership tend to depend on other clusters. As in paper citations, where the records contain attributes about authors, paper, conference, and venue, it is often necessary that two author attribute mentions refer to the same author when the paper titles match or when the conferences are same. Similarly two paper citations may refer to the same paper despite the variations in the titles when the authors and conference attributes match. The proposed ER techniques uses one of these approaches for cluster-membership decisions: a) Non-probabilistic approach: *similarity propagation* [DHM05]; b) *Probabilistic approaches*: generative models [BG06] and undirected models [MW03]; and c) *Hybrid approaches* [SLD05, ARS09].

Similarly propagation algorithms work by defining a graph which encodes the similarity between entity mentions and matching decisions, and compute matching decisions by propagation similarity values. *Reference reconciliation* paper [DHM05] considers entity resolution in complex information spaces. The proposed techniques along with similarity in one class or domain, also take into account the relations of entities in other classes or domains. The proposed algorithm involves identifying associations between entities, propagating the associations discovered in order to accumulate positive and negative evidences for other entities, thus leading to further associations or dissociations of other entities. They keep track of negative constraints enforcement and fix any resulting inconsistencies. Collective relational clustering [BG07] after constructing attribute and relational similarity graph, make use of hierarchical agglomerative clustering to merge clusters of mentions. When clusters are merged, all the related clusters similarity values are updated and propagated. Domain independent data cleaning [KM06] makes use of blocking for finding the initial bootstrap clusters. They repeatedly find closest cluster pair and merge them to form a bigger cluster if no constraints are violated. All these similarity propagation approaches can make use of probabilistic models locally, but there is no global probabilistic model. Thus these techniques are often found to be more scalable.

Collective ER techniques based on probabilistic approaches aim to model a global probabilistic model for making the decisions. The generative models (LDA-ER [BG06], Bayesian Networks [PMM$^+$03]) aim to model dependencies between match decisions in a generative manner. The disadvantage of generative models is that it requires the underlying similarity graph to be acyclic. Undirected probabilistic approaches rely on Markov networks based probabilistic semantics. Many of the traditional ER techniques make the assumption that two records are pairwise independent, and fail to exploit the correlations for identifying the further resolutions. The paper [MW03], foregoing this assumption, makes use of undirected graphical models to introduce several discriminative, conditional probability models for entity resolution. The conditional models help in incorporating a great variety of features of input without being concerned about their dependencies. Other examples of undirected probabilistic approaches can be seen in Markov Logic Networks based ER [SD06], where in the constraints can be defined declaratively based on first order logic syntax; and probabilistic similarity logic based ER (PSL) [BMG10].

Constraint-based ER approaches explicitly encode relational constraints. *Hybrid approaches* help in formulating these constraints as an hybrid of constraints and probabilistic models or as a constraint optimizing problem. Constraint-based Entity Matching [SLD05] and Dedupalog [ARS09] are two examples where the constraints are specified as probabilistic graphical models. In summary, the similarity propagation approaches often scale better than the probabilistic models but are often cumbersome to specify. The probabilistic models are often expensive but easier to specify. Making these techniques scalable is an active research area.

2.2.2 Constraints based ER

Constraints play an important role in the ER. They not only help in reducing the complexity, but also avoid inconsistencies. We explain some of the important forms of constraints with which ER algorithms deal below.

Consider M_1, M_2, M_3, and M_4 to be entity mentions in a dataset, which may or may not refer to the same entity.

1. **Transitivity**: If M_1 and M_2 match, M_2 and M_3 match, then M_1 and M_3 match. Similarly, if M_1 and M_2 match, M_2 and M_3 do not match, then M_1 and M_3 do not match.

2. **Exclusivity**: If M_1 and M_2 match, then M_2 and M_3 cannot match. Likewise, if M_1 and M_2 do not match, then M_2 and M_3 can match.

3. **Functional Dependency**: When M_1 and M_2 match, then M_3 and M_4 must match. Similarly, if M_1 and M_2 do not match, then M_3 and M_4 cannot match.

The above discussed are generic constraints. However, there could be many domain specific constraints. Constraint ER paper [SLD05] discusses a number of semantic integrity constraints specific to ER in paper citations records. Some such semantic constraints include: *Aggregate Constraint:* No researcher has published more than five AAAI papers in a year; *Incompatible constraint:* No researcher exists who has published both HCI and numerical analysis; *Ordering Constraint:* If two citations match, then their authors will be matched in order; etc.

In [Fan08], the authors propose a class of integrity constraints for relational databases, referred to as conditional functional dependencies (CFDs), and study their applications in data cleaning. In contrast to traditional functional dependencies (FDs) that were developed mainly for schema design, CFDs aim at capturing the consistency of data by enforcing bindings of semantically related values. Their work not only yields a constraint theory for CFDs but is also a step toward a practical constraint-based method for improving data quality.

Chauduri et al. in [CSGK07] show that aggregate constraints (as opposed to pairwise constraints) that often arise when integrating multiple sources of data, can be leveraged to enhance the quality of deduplication. However, despite its appeal, they show that the problem is challenging, both semantically and computationally. By defining a restricted search space for deduplication that is intuitive in the context, they solve the problem optimally for the restricted space. Their experiments on real data show that incorporating aggregate constraints significantly enhances the accuracy of deduplication.

Robust identification of fuzzy duplicates [ACG02, CGM05] proposes a new formulation for duplicate elimination problem based on two properties namely: *Compact Set* (CS) and *Sparse Neighborhood* (SN), that characterize the duplicate tuples. It is intuitive to add additional constraint predicates to their framework. They show that their formulation has several desirable characteristics under intuitive transformations to distances between tuples.

Probabilistic approaches discussed above in Collective ER Section 2.2.1 are also capable of modeling and propagating the constraints. Constraint-based entity matching [SLD05] describes a probabilistic solution that exploits integrity constraints that frequently exist in the domains, to improve the matching accuracy. The paper describes a novel combination of EM and relaxation labeling algorithms that efficiently learns the generative model, thereby matching entities in an unsupervised way. Correlation clustering techniques [ACN08] address optimization problems in which contradictory pieces of input information are given and the goal is to find a globally consistent solution that minimizes the extent of disagreement with the respective inputs.

2.2.3 Big Data and Distributed ER

When matching records from two databases, one approach needs each record from one database be compared with all records in the other database in order to determine if a pair of records corresponds to the same entity or not. When de-duplicating a single database, each record potentially needs to be compared with all others. The computation complexity of data matching therefore grows quadratically, i.e. $O(N^2)$, as the databases to be matched get larger. On the other hand, the number of potential true matches (i.e. pairs or groups of records that refer to the same entity) only grows linearly, i.e. $O(N)$, with the size of the databases to be matched. If it is assumed that the databases to be matched do not contain duplicate records, then the maximum possible number of true matches is limited by the size of the smaller of the two databases.

The second approach of keeping computational costs low becomes critical when dealing with large data sizes. This computational challenge is addressed by techniques like *blocking or canopy generation*, that aim to efficiently and effectively remove record pairs that likely do not refer to matches, while selecting candidate record pairs for detailed comparison and classification that likely will be matches. For example, when we look into customer records in a database, it is mostly likely that customer records with different cities will not refer to the same customer. In such a case an efficient blocking technique would be city-based. Only records falling in the same canopy (*blocking criterion: city*) need to be compared, where-by avoiding comparing records that fall in different canopies, thus reducing computational cost significantly. Examples of simple blocking keys include: first four characters of last name, $City + State + Zip$, ngrams, etc. More complex blocking functions are explored as: conjunction of simple functions [MK06, BKM06], chain trees and blkTrees [DSJMB12].

Hash-based blocking works by assigning a hash key (h_i) to each canopy (C_i). A record r is assigned to C_i if $hash(r)$=h_i. So, each hash value results in disjoint blocks. All pairs within a block are compared, while pairs across canopies are never compared, thus keeping computational costs low. The hash functions could be deterministic function of attribute value or combination of attribute values. They could also be boolean functions over attribute values as used in [BKM06, MK06, DSJMB12]. Another popular technique is $MinHash$ technique [BCFM98]. MinHash or the min-wise independent permutations locality sensitive hashing scheme is a technique for quickly estimating how similar two sets are. It has been used in search engines to detect duplicate web pages and eliminate them from search results and also been used in large-scale clustering problems, such as clustering documents by the similarity of their sets of words.

Pairwise similarity or nearest neighborhood is one other way of blocking, where in nodes (records) according to similarity metric are clustered together and grouped into non-disjoint canopies. The merge-purge problem in [HS95], which is about merging data (large scale) from multiple sources in as efficient manner as possible, while maximizing the accuracy, makes use of sorting for ordering all its records, by which similar records fall in the same neighborhood, followed by clustering techniques for finding the canopies. The sorted neighborhood is very expensive because of sort step involved. While clustering is good but does not have high accuracy. The authors propose a novel approach called multi-pass approach which performs well computationally and achieves better accuracy. In this approach, merge-purge process is done multiple times over small windows followed by the computation of transitive closure.

Canopy clustering is one other blocking technique. Applying exact clustering techniques on the complete dataset is expensive. Using canopy clustering the data sets are cheaply partitioned in to approximate overlapping subsets (canopies) and the exact clustering techniques are now applied on the canopies, thus keeping computational cost low. The paper [MNU00] proposes efficient clustering techniques even when the datasets are huge, have high dimensionality and the target number of clusters are huge. The proposed technique involves two steps. In the step one, the data is partitioned into `canopies` based on a simple metric (which is computationally less intensive). The so formed canopies can be overlapping. In the second step, more sophisticated metrics can be used to cluster the data in the canopies independently. The important gain in this approach is that data points across canopies need not be compared, thus reducing the computation cost significantly.

ER for big-data efforts include extending the existing algorithms or creating new ones that could be done in a distributed manner. MapReduce [DG04], proposed by Google and popularized by Hadoop [Whi09] community, is a simple programming model for processing large datasets with a parallel, distributed, and fault tolerant algorithm on a cluster. The large datasets are distributed across cluster of nodes. MapReduce framework is efficient for tasks that can perform computations locally on the nodes and keep data exchange across nodes to the minimum. MapReduce involves two important phases on Map and Reduce. Distributed ER techniques work well with disjoint blocking techniques, such as: hash-based blocking [VCL10], distance-based canopy clustering on map-reduce [Mah], iterative blocking [WMK$^+$09]. These disjoint blocking techniques can be implemented in Map phase and are useful in localizing the blocks; however, the remainder of the ER algorithm that is implemented in Reduce phase, needs information from multiple reducers. Computing connected components among canopies [KTF09, RMCS12] with message passing [RDG11] in addition to blocking aid in realize distributed ER.

Recent efforts in ER have been extending the algorithms to distributed systems. IdMesh [CM07] tackles the problem of managing identities on the web. The authors describe a decentralized infrastructure supporting efficient and scalable identity management and demonstrate the practicability of their approach in a deployment over several hundreds of machines.

2.2.4 Entity Resolution for Web Documents

We have seen a number of techniques that address the ER problem in databases and citations domains. As we concern ourselves with solving Entity Resolution problem for web document collections, the ER techniques proposed earlier for DB and citations domain do not apply readily. Web document collections pose a number of challenges, mainly because the web data , we are interested in, is unstructured. As most of the proposed ER techniques work on structured records, they fall short when they are applied to web documents. In this section, we look into number of research efforts addressing the ER problem for web based records.

The paper [MBGM06] presents a pairwise comparison-based method, where the authors consider confidence values during the resolution process. They propose to merge database records, which refer to the same entity, right away, as they are found to be equivalent by the algorithm. The algorithm also computes a new combined confidence value for the merged record. A more complete analysis of results can be found in [BGMM$^+$09], where the authors also study, how to chose the sequence of the records to be processed, such that the running time of the algorithm remains low. Chauduri et al. [CGM05] introduce a model for detecting fuzzy duplicates in databases. They extended their model also to a more general setting in [CSGK07]. Their paper is particularly important from methodological point of view, as they systematically derive their entity resolution algorithms from an axiomatic model. Unfortunately their model cannot be easily extended to the Web context because the properties of similarity functions for entities in Web documents do not show the same properties as in the case of fuzzy duplicates, so the basic assumptions of their model are not satisfied.

A number of commercial tools are available for duplicate records elimination in databases domain. Some examples include: SQL Server Tools[3] (Microsoft), DataBlade[4] (IBM), ETI* DataCleaner[5] (ETI), Trillium[6], WizRule[7] (WizSoft), ChoiceMaker[8], etc. Refer to [BG05] for an extensive survey on data quality commercial tools. As these tools are developed in the context of (relational) databases, they are ill-equipped to deal with the similar problem in web documents collection.

Kalashnikov et al. [KM06] study Entity Resolution in Web context. They propose to create an entity resolution graph using the feature-based similarities. The graph witnesses the uncertainty of the features by having multiple nodes, the so called "*choice nodes*" are corresponding to possible references to a given entity. The authors apply heuristic graph measures to measure the connectedness of entities. The underlying idea behind their heuristic is the "*context attraction principle*": if two entities are related

[3] http://www.microsoft.com/en-us/sqlserver/default.aspx

[4] http://www.ibm.com/software/data/informix/

[5] http://www.eti.com/

[6] http://www.trilliumsoftware.com/

[7] http://www.wizsoft.com/

[8] http://sourceforge.net/projects/oscmt/

then it is likely that there are multiple chains in the entity resolution graph between their corresponding nodes. The authors further improved their techniques in [KCMN08]. In [KCMN08] and in many other approaches, such as for example in [DHM05], the authors consider a more complex graph, which captures more complex relations rather than the similarities between the entities as in our work. In this thesis, as we show in Chapter 3, we limited ourselves to a simple representation and to focus the issues in this simpler case, our framework could be later extended to a more complex setting. Their work and their use of context information in [KM06] is a similar technique to our quality-aware similarity assessment technique. We rely on different features, which are also easier to estimate.

IdMesh authors Cudré-Mauroux et al. [CMHJ$^+$09] take a different approach to entity resolution in the Web context. They propose a graphical model-based probabilistic framework to capture the relations among the entities. Their framework also includes trust assessments about the providers of the entity equivalence assertions. These trust assessment values are later adjusted as their probabilistic reasoning framework eliminates the detected inconsistencies. While this approach has many advantages, it is not fully applicable to our case, as the underlying factor graph model would have very large cliques, as subgraphs, which could easily lead to poor convergence of the probabilistic reasoning.

In certain cases, person names appearing on Web pages might be annotated with a globally accepted ontology. This direct link between the person names and the ontology helps to disambiguate the person names. However, such globally accepted ontologies are not present in the emerging Semantic Web. Instead, ontologies are very often used as local schema, thus one needs to relate the existing annotation to an ontology one would like to use. The Semantic Web community has developed a plethora of such techniques, see [ES07]. The OKKAM project suggests a different approach [BPSV09]; they propose an Entity Naming Service (ENS), which provides globally unique identifiers for entities on large scale, for (Semantic) web applications. Their approach relies on the existence of a large and clean (i.e. resolved) collection of entity profiles. Entity profiles collect relevant attributes of real world entities. Our techniques proposed in this thesis can contribute to create or extend such an entity profile collection.

Combining classifiers

Many of the ER techniques relevant to web domain make use of multiple classifiers that are based on multi-set features; they also need to employ sophisticated techniques to combine the decisions from such multiple classifiers. Combining multiple classifiers is studied extensively in the machine learning and data mining communities [SE10]. We make use of such techniques in Chapter 3 when addressing the ER problem for Web documents. In summary, these techniques can be broadly divided into two main categories:

1. *Classifiers Fusion*: In which the final decision on a sample point is based on the fusion of decisions of individual classifiers, in some sense similar to achieving consensus. Examples include majority voting, weighted voting.

2. *Dynamic Classifier Selection*: In this scenario, the decision of one of the classifier is chosen as the combined decision. Here, the classifier is chosen based on which classifier best represents the sample point.

The combination of classifiers rely on dividing the sample space into regions, and estimating the accuracy of the participating classifiers in each of the sample regions. Various works like [WKB97, LY01, SGC02, CKM09, BM03, ZR05, BGB08] propose different ways of defining these regions, and present simple to sophisticated ways of combining the decisions of various participating classifiers. We discuss their approaches in detail and contrast our work in Section 3.6.

2.3 Entities in Twitter like Micro-blogging Platforms

Twitter kind of microblogging services allow people to publish, share and discuss short messages on the Web. On average, Twitter users publish more than few hundred million tweets per day[9]. Given the tremendous growth of such microblogging platforms in the recent years, it has undoubtedly attracted great interest from both industry and academia. Many public and private organizations have started to monitor Twitter streams to collect and understand users' opinions about the organization entities. With the popularity of social networks, where people express themselves on these networks, many organizations, sports teams, TV shows, etc., are interested in mining these networks and provide real-time social pulse related to their products.

Given the huge amount of data that is produced on these social networks, it is essential to correctly identify subset of the data that is relevant to the entity one is interested. Even though this data is publicly accessible, nevertheless, the noisy and short-context-less nature of the tweets brings in new challenges. The ER techniques discussed earlier perform poorly when applied directly on Twitter kind of data. In the following Section 2.3.1, we see number of research works that are based on mining of the Twitter data. Next in section Section 2.3.2, we discuss the efforts concerning extraction of useful information (sentiment metrics) from social networks like Twitter. Finally in Section 2.3.3, we present works that have explored entity related classification of Twitter data.

2.3.1 Mining of Twitter Data

Twitter has seen exploratory growth in last few years. Due to which the Twitter data is of interest in many research works for a number of reasons. While some works [ZJW$^+$11, GAHY12, JZSC09] were interested in understanding how this new media is in comparison to the other forms of existing media (news channels, news websites, etc) and social networks, others [GAC$^+$10] were interested in modeling information propagation and temporal dynamics in this new medium. Some other works [WHMW11, AGHT11, HMOS12] were interested in understanding users' behavior and usage patterns in Twitter, while some others [SST$^+$09, PP10] were interested in assessing the sentiments and opinions of the Twitter users. Other class of works include adapting tools (extracting entities, inferring topics, etc.) that are applicable to document collections to twitter kind of data.

One of the challenges we deal in this thesis work is the task of Entity Matching of tweets, where we are interested in classifying a tweet message with respect to an entity (see Chapter 4). Also at the core of many of the above works, lies the problem of classifying tweets with respect to a criteria. We give brief overview of such works which need to address the classification of tweets problem. Some of the relevant works include [SFD$^+$10], TwitterStand: news in tweets [SST$^+$09], Twitter as corpus [PP10], tweets as electronic word of mouth [JZSC09], etc.

[9] http://articles.washingtonpost.com/2013-03-21/business/37889387_1_tweets-jack-dorsey-twitter

The authors of information filtering in Twitter [SFD+10], take up the task of classifying the tweets from twitter into predefined set of generic categories such as News, Events, Opinions, Deals and Private Messages. They propose to use a small set of domain-specific features extracted from the tweets and the user's profile. The features of each category are learned from the training set.

In TwitterStand [SST+09], the authors have built a news processing system based on Twitter. They built a system that identifies messages from Twitter streams that correspond to late breaking news. Some of the issues they deal with are: separating the noise from valid tweets, forming tweet clusters of interest, and identifying the relevant locations associated with the tweets. All these tasks are done in an online manner. They also build a Naive Bayes classifier for distinguishing relevant news tweets from irrelevant ones. They represent intermediate clusters as a feature vector, and associate an incoming tweet with cluster if the distance metric to a cluster is less than a given threshold.

Discovering of geographical topics from Twitter streams is explored in [HAG+12]. The work proposes a unified model for diversity in twitter considering the topical diversity, geographical diversity and user interest diversity. They make use of sparse generative techniques for the unified model. Using this unified model they are able to predict accurately the geographical location based on the tweet message. They are also able to uncover the topics for different locations. The paper [TKW10] proposes a technique to retrieve photos of named entities with high precision, high recall and diversity. The innovation used is query expansion, and aggregate rankings of the query results. Query expansion is done by using the meta information available in the entity description. The query expansion technique is very relevant for our work, it could be used for better our entity profile creation techniques, which we present in Chapter 4.

The authors of short messages clustering [PTPCR11], address the problem of company identification in the micro-blogs by resorting to clustering techniques as a parallel approach to designing classifiers. They propose techniques to improve the representation of a twitter message through term expansion, in a process to enrich the semantic similarity hidden behind the lexical structure.

Identifying relevant tweets for Social TV [DFD11] look into similar problem –classifying tweets with respect to an entity– in a different setting. They address the problem of filtering twitter messages for Social TV purposes. They are concerned if a tweet message is about some popular TV show (Lost, Survivor, Friends etc). Their approach, somewhat similar to the approach we propose in Chapter 4, is of bootstrapping a model with smaller training set, developing a more sophisticated model using large dataset of unlabeled messages and further using domain specific features to obtain a final classifier. However, their focus was on developing a generic classifier that can be used on any unseen TV show in the training set.

2.3.2 Sentiment Extraction from Twitter Data

Sentiment analysis and opinion mining based on the user generated content is studied in various research efforts [DPH+09, Tur02, PLV02, HL04, GSS07]. User written reviews based sentiment analysis can be seen in [Tur02, PLV02, HL04]. Authors of *product-reviews sentiment mining* [HL04] perform feature set based sentiment analysis. They extract product specific features from the review texts using Noun-Noun phrases, and compute sentiment metric along each feature. While the product-reviews work [Tur02] uses adjectives and adverbs for performing opinion classification of the product reviews. They employ Pointwise Mutual Information - Information Retrieval (PMI-IR) algorithm for estimating the sentiment orientation of the phrases. Where as in [PLV02], the authors tested various machine learning algorithms on Movie Reviews. In [DPH$^+$09], Dray et al. instead of generic adjective based sentiment analysis, they made use of domain specific adjectives to perform sentiment analysis. They observed that predefined lexicon fails to capture domain specific information. Each blog-post is classified as positive, negative or neutral using the classifier that is built for the domain of the blog post. Godbole et al. [GSS07] present a system that assigns scores indicating positive or negative opinion to each distinct entity in the text corpus. Their system consists of a sentiment identification phase, which associates expressed opinions with each relevant entity, and a sentiment aggregation and scoring phase, which scores each entity relative to others in the same class. Finally, they evaluate the significance of their scoring techniques over large corpus of news and blogs.

All of these promising works deal with larger texts, as they employ NLP tools developed for larger text documents collections. They do not perform well when applied to shorter text messages like tweets. Here we look into few works that performed sentiment analysis of Twitter data.

Mood analysis on tweets

Social-network services facilitate users to share their ideas, opinions, pictures, videos, news, and other various forms of contents in the Web. Such social data typically contains highly valuable information, aiding a wide range of applications; for example, allowing social scientists to understand human behavior, companies to figure out their customers' preferences, news agencies to identify significant news, political analysts understanding the political pulse of the nation, etc. Previously, it was difficult to obtain the rich set of social information, or required large amounts of laborious human efforts like conducting surveys, interacting with the users. With the advent of Web 2.0, all this information is readily available, leading to a variety of interesting research works.

One popular research line is to extract and analyze mood information from Twitter messages [MBB$^+$11, BMZ10, Pul, TBP11, PP10, AXV$^+$11]. In [MBB$^+$11] micro-blogs are used for mood analysis, where they present a method for associating mood to certain events. Their techniques help in summarizing huge volumes of tweets w.r.t. the events. The TwitInfo system proposed by the authors, allows users to browse a large collection of tweets using a timeline-based display that highlights peaks of high tweet activity corresponding to the events. Similarly, the authors of *Pulse of Nation* [Pul] by extracting sentiment information from Twitter messages are able to track the national mood. This study analyzed over 300 million tweets corresponding to the US region over a period of 3 years . They present the moods across the country using different cartograms; and observe the variation of nation's mood over 24-hour period of a day and the days of a week.

Another study [BMZ10] tries to predict the impact of public mood expressed in Twitter messages on the stock market, by investigating the correlation of moods inferred from large-scale twitter feeds with the Dow Jones Industrial Average. They make use of mood tracking tools, namely, OpinionFinder (that measures positive vs. negative mood) and Google-Profile of Mood States (GoPMS) that measures mood in terms of 6 dimensions.

The authors of [TBP11] analyze Twitter messages in order to study why certain events resonate well with the population. They assess whether surges of interest in Twitter are associated with heightened emotions, by checking if the average sentiment strength of popular Twitter events is higher than the Twitter average, or by assessing whether an important event within a broad topic is associated with increased sentiment strength.

Research works [PP10, JZSC09] make use of Twitter for the task of sentiment analysis. They build a sentiment classifier based on a tweet corpus. Their classifier is able to classify tweets as positive, negative, or neutral sentiments. The papers identify relevant features (presence of emoticons, n-grams, hash-tags), and train the classifier on an annotated training set.

[TSSW10, CWS12] use Twitter data, to study the political inclination of the crowd in order to predict the outcome of US presidential elections. Where as in [AXV$^+$11], the authors present a method for tweet sentiment identification using a corpus of manually pre-annotated tweets. They also present a sentiment scoring function which uses prior information to classify and weight various sentiment bearing words/phrases in tweets.

We will show in Chapter 4 that these works are complementary to ours. Our techniques, which identify the tweets relevant to an entity, could serve as an essential preprocessing step to these sentiment or opinion analysis based works. The works we discussed in this section, which aid in sentiment extraction from tweet messages, present tools & techniques that are useful in our social metric extraction task of data fusion process (Chapter 6).

2.3.3 Entity-based Classification of Tweets: Approaches

Many works based on entity identification and extraction, for example in [BM05], [CKM09], [KCMN08], [YMA10b], usually make use of the rich context around the entity reference for deciding if the reference relates to the entity. However, in the current work (entity-based classification of tweets), the tweets which contain the entity references usually have very little context, because of the size-restrictions of tweet messages. Our work addresses these issues, namely how to identify an entity in scenarios where there is very little contextual information.

It is also common practice to use hash-tags (for example #apple) in the tweet messages, when users intend to refer to a particular entity (#apple refers to Apple Inc. company entity). Facebook also introduces hash-tags[10] for linking topics and events discussed by its users. Works [LWH$^+$12, MWL$^+$12] have relied on hash-tags to filter-out messages corresponding to a particular entity. While this applies to popular entities, it is not possible to use it for every possible entity. Also, not every tweet message about the company entity makes use of the hash-tag, resulting in the system missing out many relevant tweets. In this thesis work, we want to identify all tweets relevant to an entity, irrespective of presence of hash-tags in the messages.

We summarize the different classifiers [YMO$^+$10, Kal10, CVSPO10, TB10] proposed by various research groups for the WePS-3 challenge task [AAG$^+$10], of classifying tweets based on entity. The ITC-UT system [YMO$^+$10] was built according to rules based on Part of Speech tagging and Named Entity extraction. The system –by considering the linguistic aspect of the company mentions– achieves acceptable accuracy. The classifier realized in the SINAI system [CVSPO10] makes use of Named Entity extraction from the tweet messages. The performance of the classifier varied across various companies. It is difficult to predict for what kind of companies this classifier performs well. From the above two systems it can be seen that Named Entity extraction does bring in some accuracy, but these tools are not designed for short and context-less messages like tweets. KAMLAR systems [Kal10] build their classifier starting with a bootstrapping step based on the vocabulary of the home page. This system –even though it has low on overall accuracy– had decent F-score for relevant tweets, suggesting that

[10] http://abcnews.go.com/Technology/wireStory/facebook-introduces-hashtags-19384181#.Ubi3AVltg_g

a bootstrapping step can be very useful for company names with high ambiguity. Another approach described in [TB10], focuses on working with organization independent features and not relying on any external information sources. Their approach of using J48 decision tree classifier is quite interesting, but the drawback it relies heavily on the availability and quality of the training set.

In Chapter 4, we present our basic profile classifier (LSIR-EPFL classifier [YMA10a]) that was the winner of WePS-3 evaluation challenge. The LSIR-EPFL classifier essentially makes use of different information sources on the Web to create an entity profile. These semi-automatically created company profiles are essential for accurately classifying the tweets based on the company entity. We further extended the basic techniques with Active Learning [YMA11], through constant monitoring of the Twitter streams. We also discuss further details on the work and introduce systematic performance analysis in Section 4.6.

2.4 Entity Profiles

As regular users of the social networks, people share and communicate their thoughts and opinions via Facebook, Twitter, or other numerous social platforms. The topics discussed on these media is of diverse nature ranging from user specific topics, news to casual chatter. This large reservoir of social data is of great benefit to applications – in particular for those that rely on information about its users. A specific user entity publishes content on these platforms, which reflect his interests, personality, expertise, etc. Entity profiling tasks concerns about creating compact summary of an entity (e.g. an user entity) based on the content related to the entity. In this section we review different works that deal with extraction of topics from content, which eventually aid in constructing entity profiles.

Probabilistic modeling (LSI[11], pLSI[12], graphical models [BNJ03, RHNM09], etc) has been extensively used for discovering latent topic structures in data in text documents. Topic discovery, topic evolution, document classification and clustering, etc., are some of the problems studied for modeling and profiling of knowledge in text documents collections. These works are essential in summarizing huge document collections, helping in exploring the collections, retrieving documents that are semantically relevant to the queries, and etc. Topic models [JRT10] and language modeling [BAdR06] is used in identifying topic based experts in enterprise document collections.

2.4.1 Topic Modeling in Micro-blogging Platforms

As we are interested in profiling an user entity and a location entity based on the Twitter and other social networks content related to the entity, we look into different research works that apply topic modeling to Twitter content in this section.

A number of recent works have explored the use of topic models in the Twitter domain for modeling Twitter messages and users [HD10], finding topical authorities [PC11, WLJH10, ZTL07], making recommendations [HBS10], and comparing it with other media [GAHY12, ZJW^{+}11]. We also focus our attention on works that have explored user modeling [AGHT11, GAHY12, HMOS12, AHK11] in micro-blogging platforms.

[11] http://en.wikipedia.org/wiki/Latent_semantic_indexing

[12] http://en.wikipedia.org/wiki/Probabilistic_latent_semantic_analysis

Works like [LWH$^+$12, RCME11] have focused on adapting techniques and tools that were successful on text corpora to the recent vastly popular micro-blogging platforms. They adapted the named entity extraction (NER) techniques for the shorter and noisy micro-blog posts. The NER task is a critical step for the task of identifying the subset of tweets that are relevant to an entity which we tackle in Chapter 5 of this thesis.

Topic modeling of Twitter messages has been considered in [HD10], where models for three different tweet aggregation strategies have been considered: First, each Twitter message is considered as a document; second, all the tweets corresponding to a user are considered as being a single document; and finally, all tweets containing a particular term are put together in a one single document. These three strategies are referred to as MSG-Topic-Model, USR-Topic-Model and TERM-topic model. Each document D is considered to be sampled from a topic distribution (θ), and each topic has ϕ distribution over the words. The documents are generated based on the θ and ϕ distributions. One uses Gibbs Sampling to estimate the values of θ and ϕ. They show that the topics learned by the various schemes are different in quality. The topic models learned from aggregated messages of a user can lead to superior performance in classification problems. Based on their study, in our current work we grouped all the tweets corresponding to a user in to a single document and used it to infer the users' topics.

Several previous works [PC11, WLJH10, ZTL07, RDL10] have used topical modeling features on micro-blogging platforms for finding topic-based experts and authorities. The authors in their work on topical authorities in microblogs [PC11] propose various sets of features in order to find topic-based authoritative users. The set of features are based on how frequently users tweet, what percentage of their tweets are retweets, how often their tweets are retweeted, how often users are mentioned by other users, and how diverse or focused are the tweets to a particular topic. TwitterRank [WLJH10] proposes a ranking algorithm, which is an adaptation of PageRank algorithm, for finding topic-sensitive influential users. They make use of LDA on the twitter content for linking an user with certain set of topics, and use topic level similarity among users as feature of their ranking algorithm.

Expert finding in Social Network, combines personal local information with network information to find the experts on a topic. The approach proposed in [ZTL07] involves two steps: initialization and propagation. The initialization step forms an expert profile just based on the local information, and a propagation model is applied in the next step in which expert scores from one node are propagated to the neighboring nodes. Such approaches could be combined with the ones we propose in this thesis work to improve the quality of both tweet disambiguation as well as of expert finding.

Most user interactions in Twitter are still primarily focused on the social graphs. Characterizing micro-blogs with topic models [RDL10] explores content analysis of Twitter feeds for addressing special information needs of the users. They apply LDA [BNJ03] and labeled LDA [RHNM09] for identifying the latent topics of Twitter messages. Using unsupervised LDA they assign latent topics into one of the four subcategories {*substance*, *social*, *status*, and *style*}. The partially supervised labeled LDA could assign labels (emoticons, hashtags, etc.) to the latent topics extracted from the Twitter feeds. We apply similar techniques for the problem of tweet disambiguation.

Some works, as in [ZJW$^+$11, GAHY12], have relied on topic modeling for comparing recent micro-blogging platforms and traditional news media platforms. In the paper [ZJW$^+$11], the authors do an empirical comparison of the Twitter content with that published on tradition media like the New York Times. Using standard LDA they infer topics from the news dataset, while they propose a Twitter-LDA model for extracting topics from Twitter data. This study shows how certain topics are popular on Twitter while some others are popular on news media. In [GAHY12] the authors extend their user modeling framework [AGHT11] for comparing the usage behavior on two popular micro-blogging platforms: Sina Weibo[13] and Twitter.

In [KML13, WC10] the authors present LDA transfer learning. Transfer Learning is the process of generic learning in one domain and applying the model in a different domain. In topic-bridged LDA ($tLDA$) a model is built from a variety of labeled and unlabeled documents, and they apply transfer learning for document classification task.

2.4.2 User Modeling over Micro-blogging platforms

Web is gradually transforming itself as a users personal archive, where users not only find information but leave, share and archive information [LMB$^+$13]. Twitter being widely adopted, real time and representative of the users, despite being of noisy nature, is a great source for modeling a user [YMH$^+$]. User profiles were constructed in [SCS09, AGHT11, HMOS12] for better news and people-to-follow recommendations, dealing with information overload, understanding users' expertise and interests, etc. [SCS09] make use of entity profiles, that are sets of information extracted for each ambiguous person in the entire document, and features based on topic models to cluster documents – containing a person name – based on the actual person entity. Authors of [AGHT11] analyze user modeling on Twitter for personalized news recommendations. Their framework helps in creating user profiles that are based on extracted topics and entities from the tweet content, and show its superior performance compared to hash-tag based user profiles. They also consider temporal aspects of the user profile for better news recommendations.

The work [HMOS12] proposes techniques to construct multi-faceted user profiles for Twitter users, thereby helping one to navigate the complex domain-space represented by Twitter. Their model profiles users and their social networks using tags and labels from curated lists. In our future work, we plan to make use of the user maintained lists and the lists to which an user belongs in improving the quality of our constructed user profiles. [AHK11] work extracts professional interests from social web (Facebook, Twitter) profiles. Twittomender [HBS10] explores building of user profiles based on tweets which are grouped as users' own tweets, followers tweets and followees tweets. They make use of TF-IDF ranking technique in construction of the user profile, which they use for recommending other Twitter users to follow.

We present our techniques to construct user and location entity profiles in Chapters 5 & 6. We also present few applications that are based on such generated entity profiles, namely, making sense of microposts by using the user entity profile as an additional context to the messages and user-based travel plan recommendation system.

2.5 Summary

In this chapter, we presented a comprehensive overview of research works that addressed various entity-related challenges. Here, we surveyed a number of techniques that addressed ER problems under various domains. We started with the solutions proposed for traditional DB domain, citations domain, and to

[13] http://www.weibo.com

the ways it was addressed for web documents. We broadly grouped these techniques under *pairwise and collective ER*, *constraint-based ER*, and *distributed ER*. In this thesis work, specifically in Chapter 3, we are interested in solving ER for web documents. Most of the ER methods, proposed for DB and citations domain, are primarily designed for structured records. They fall short when applied to unstructured web documents. We propose a generic framework for solving ER for web documents in the next chapter.

In the second part, we presented a number of works that have relied on mining the Twitter data. In this thesis we are interested in entity matching in microblogging environments. We designed our techniques partly based on number of features extracted from these Twitter data works. It is also important to realize that entity-matching methods rely on the context surrounding the entities. Entity-matching in Twitter environments provides additional challenge of dealing with noisy and short context-less tweet messages. In Chapter 4, we present our entity-based classification of tweets techniques that overcome the challenges posed by shorter texts. We have also seen several works that dealt with sentiment extraction from tweet messages. The sentiment extraction techniques can be applied on relevant subset of tweets, which can be identified using our proposed techniques.

User modeling has been studied extensively to understand user preferences and interests. Also a number of sophisticated approaches have been proposed to extract concepts and topics from text documents. In the final part of this chapter, we gave an overview of works that focused on extracting topics from Twitter data. We adapt these techniques for profiling user and location entities based on the content related to the corresponding entities in Chapters 5 and 6. We also present applications that rely on these entity profiles.

Part II

Entities in Web Documents

Chapter 3

Entity Resolution for Web Documents

Entities should not be multiplied beyond necessity. (14^{th} century).

Entity identifiers should not be multiplied beyond necessity (21^{st} century).

OKKAM's razor principle, Then and Now

One of the key challenges to realize automated processing of the information on the Web, which is the central goal of the Semantic Web, is related to the entity resolution problem. There are a number of tools that reliably recognize named entities, such as persons, companies, geographic locations, in Web documents. The names of these extracted entities are however non-unique; the same name on different Web pages might or might not refer to the same entity. The entity resolution problem concerns of identifying the entities, which are referring to the same real-world entity. This problem is very similar to the entity resolution problem studied in relational databases, however there are also several differences. Most importantly Web pages, often only contain partial or incomplete information about the entities.

Similarity functions try to capture the degree of belief about the equivalence of two entities, thus they play a crucial role in entity resolution. The accuracy of the similarity functions highly depends on the applied assessment techniques, but also on some specific features of the entities. In this chapter, we propose systematic design strategies for combined similarity functions in this context. Our method relies on the combination of multiple evidences, with the help of estimated quality of the individual similarity values and with particular attention to missing information that is common in Web context. We study the effectiveness of our method in two specific instances of the general entity resolution problem, namely *the person name disambiguation* and *the Twitter message classification* problem. In both cases, using our techniques in a very simple algorithmic framework, we obtained better results than the state-of-the-art methods.

3.1 Introduction

Entity resolution is a well studied problem in the context of relational databases [FS69, HS95, CGM05, CKM07, MBGM06, DHM05, BGMM+09, IVE07]. Even if the papers are dated back quite early, this topic has also regained in importance recently. It is more and more common and easy to combine independent data sources, especially on the Web. There is a number of tools which recognize named entities, such as persons, companies, geographic locations, in Web documents. The names of the entities are however non-unique, the same name on different Web pages might or might not refer to the same entity. The entity resolution problem concerns with identifying the entities, which are referring to the same real-world entity. This problem is very similar to the entity resolution problem studied in relational databases, however there are also several differences. Most importantly Web pages, often only contain partial or incomplete information about the entities. Web pages are also much less structured as database records. Many of the models, which were developed for databases are not directly applicable in the new setting, for example the model of fuzzy duplicates [CGM05] does not fit well the new context. The information that could help here is the content of the Web pages, where the entity appears. They are on the one hand rich sources of information, but on the other hand this source is often not so straightforward to exploit, as it is very hard to distinguish the relevant information from noise and the relevant information might be even missing.

Entity resolution is essential for realizing entity-oriented view of the Semantic Web. In order to process information on Web pages automatically, one needs to identify the entities in Web documents and then match them to other entities in entity collections or to entities described by ontologies. Entity resolution is also needed to create such large entity collections themselves. This process is described in [MBB$^+$10]. Linking entities present in unstructured Web documents to each other can in many ways contribute to the development of the Semantic Web, independently of whether such large entity repositories will emerge.

We study two specific variants of the general entity resolution problem, namely the *person name disambiguation* problem and the *Twitter message classification*. In the person name disambiguation problem we are given a set of Web documents, each containing a given name, and the goal is to cluster the documents such that two documents are in the same cluster if and only if they refer to the same real-world person. In the Twitter classification problem, we are given a set of Twitter messages, each containing a particular keyword, which is a company name. The goal is to classify the messages whether they are related to the company or not. For this problem, we develop company profiles, and the task is then to match these profiles to the messages. While these problems require some specific algorithmic techniques, they both can be seen as entity resolution problems. We use these settings to demonstrate our quality-aware similarity assessment technique.

Similarity functions try to capture the degree of belief about whether two entities refer to the same real-world entity. There is a number of known techniques to derive similarity values. One can observe that the quality of these methods varies and highly depends on the input, and specific features of the input. The quality-aware similarity assessment technique combines similarity assessments from multiple sources. As opposed to other combination methods, we estimate the accuracy of individual sources for specific regions of the input (i.e. they are not global estimations) and uses this quality information to determine the similarity value. Additionally, as we are dealing with Web data, the lack of information poses an additional difficulty. We give particular attention to this challenge that is often not addressed by techniques in the machine learning literature.

We describe a systematic design of similarity assessment particularly suited for Web data, including novel ways of partitioning the input for quality-estimations. At the same time we demonstrate, that one can obtain and accuracy comparable or even better than the state-of-the-art methods with a very simple algorithmic technique, with the help of quality-aware similarity assessment. We analyze our techniques experimentally, on real-world datasets. The experiments show promising results, our error analysis shows systematic improvements. While we are studying the quality improvements within our algorithmic framework, we think that our quality-aware similarity assessment technique can lead to quality improvements in other entity resolution algorithms as well.

The rest of the chapter is organized as follows. Section 3.2 discusses the general entity resolution problem and our method of constructing quality-aware similarity functions. Section 3.3 elaborates on the person-name disambiguation problem, while Section 3.4 discusses the Twitter classification problem; both sections present algorithmic frameworks which make use of quality-aware similarity functions. Section 3.5 contains details on the experimental evaluation, Section 3.6 summarizes related work and finally we make conclusions in Section 3.7.

3.2 Quality-aware similarity assessment

3.2.1 Problem definition

We consider the following general entity resolution problem. We are given two sets of Web documents D_A and D_B (for example, Web pages, Twitter messages, semi-structured profiles), such that each document $d \in D_A$ (or $d' \in D_B$) is associated with some named entities (for example, persons, geographic locations, companies, organizations, etc.). We assume that the set of named entities is already extracted, and they are available as sets A and B (which are extracted references to the same entity type). Let R_A and R_B be the set of real world entities and for an entity $a \in A$, let $r(a) \in R_A$ denote the corresponding real world entity. The entity resolution problem aims to find the pairs (a, b), such that $a \in A$, $b \in B$ and a and b are representing the same real-world entity, i.e. $r(a) = r(b)$. Note that in some cases the set of real world entities or their relation is not known, or only partially known. In such cases, our goal is to find the pairs that best corresponds to our available training sets.

In particular, we study two specific variants of the general entity resolution, namely the *person name disambiguation* problem and the *Twitter message classification* problem. In the case of person name disambiguation problem we are given a set of documents, containing a particular name. In this setting the set D_A and D_B coincides (this is our document collection) and the goal is to cluster the set of documents, such that each document within a cluster refers to the same real-world person. In our document collection, for a given name, each document refers to only one of the persons. In other terms, there is a one-to-one correspondence between a name and a document. This assumption simplifies the algorithmic framework. Our quality-aware similarity assessment techniques are applicable also in the more complex algorithmic framework that is needed, if we drop this assumption. The number of persons (with the same name) is not known in advance. In the case of Twitter classification the set of documents D_A is a set of Tweet messages, each containing a given company name (for example, Apple). The set D_B is a set of profiles (see Section 3.4) for a given company (with the same name as D_A) and the goal is to identify whether the documents in D_A (i.e. the tweets) are really referring to the company or not, for example, decide whether the word “apple” in a tweet refers to the company Apple, represented in the profiles or something else (e.g. a fruit).

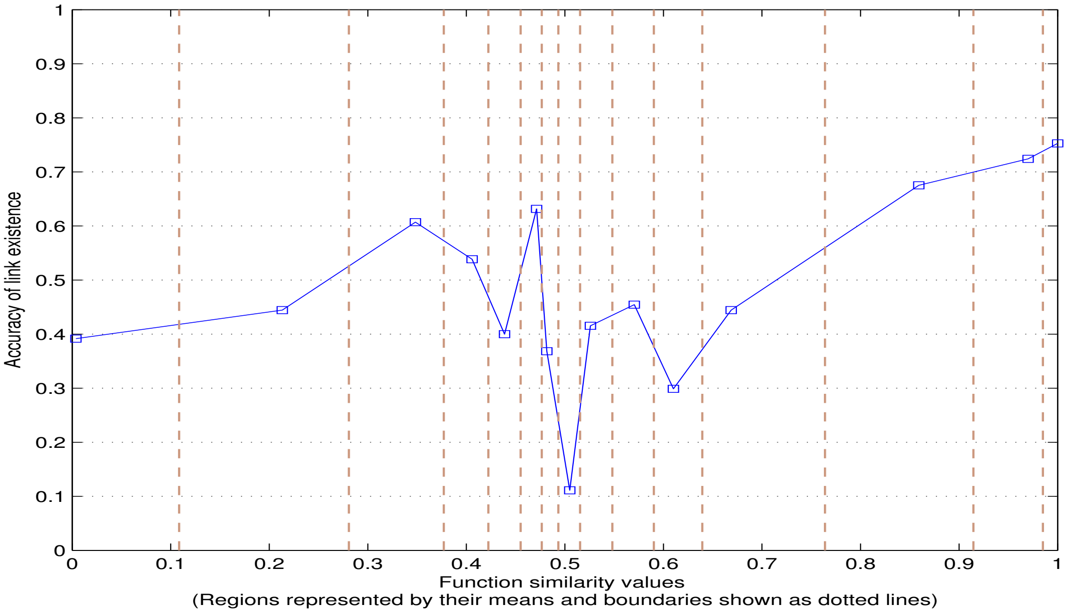


Figure 3.1: Accuracy of a similarity function

3.2.2 Challenges of assessing similarities

Assessing similarities between entities in Web documents is a challenging task. One faces (among others) the following difficulties.

- Similarity assessments focus on some specific features of the entities only. It is not clear what features one should compare and with which technique.
- Independently of which feature one chooses for assessing similarities of entities, it is likely that the Web documents contain incomplete, imprecise information about the entities or the relevant information may be completely missing. As a result, the similarity assessment techniques are often inaccurate.
- Moreover, they have varying accuracy on different input and even on different parts of the input.

In the following we give an example for the above-mentioned problem of varying accuracy, from our own experiments. Figure 3.1 shows accuracy of similarity values on a training set. On the x-axes one can see the similarity values, while on the y-axis is the accuracy of the measured value. (They are the values for the person “Cohen”, in the WWW’05 dataset, see Section 3.5.1. Even if the actual values might depend on the dataset, the variation of accuracy is a common phenomenon.) For a given interval of similarity values, we computed, how many entities match (based on the ground truth). One would expect a monotonic behavior, higher similarity values should indicate an entity match with higher accuracy.

3.2.3 Matching with quality-aware similarity assessment

We propose a technique, that addresses the above problems. While elements of this technique are known and also used elsewhere (see Section 3.6), we apply them systematically and in novel ways. As a result,

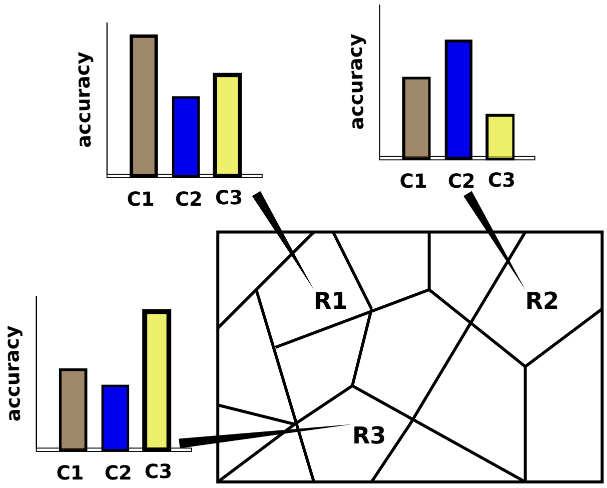


Figure 3.2: The accuracy of the similarity values varies depending on the region of the input. For example, $C1$ might have overall the best accuracy, while in region $R3$ the function with the best accuracy is $C3$.

with the help of a very simple algorithmic framework that we explain below (Algorithm 3.1) we could obtain results even better than the state-of-the-art methods.

1. We first compute similarity values, using multiple techniques, since we do not know which feature to look for. The ways we compute similarity values is specific to the particular problem, we will explain them in detail in Sections 3.3 and 3.4.

2. We partition the input into regions. In a smaller region we can much more reliable estimate the accuracy of the similarity values (Figure 3.2) than for the entire function, because each function has varying accuracy in each of these regions. In our work we used several ways to identify these regions. We explain the techniques, which are specific to the Web context, in Sections 3.3 and 3.4.

3. Using the accuracy estimations, we combine the similarity values using different combination techniques into one single similarity value, that we finally use to decide whether two entities match or not.

The simple algorithmic framework (relying on quality-aware similarity assessment) involves the following steps.

Algorithm 3.1: Generic Quality-aware Entity Resolution Algorithm

```
compute similarity values, using multiple methods
identify regions of the input, where we can estimate the quality of the computed similarity values
estimate the accuracy of each similarity value, for each region
combine the similarity values using the estimated accuracy
decide whether the entities match
output the decision
```

3.3 Person-name disambiguation

In this section we discuss the person name disambiguation problem and the use of quality-aware similarity functions for this problem. This problem is relevant for many applications, for example for person search engines who collect information from Web pages, or for news agencies (or for the online publishing industry in general). To enrich and to interlink online information (e.g. to construct `owl:sameAs` statements) person name disambiguation is essential.

Our technique relies on the quality-aware similarity assessment, and uses a simple algorithmic framework. First, in Section 3.3.1, we elaborate on the basic similarity functions we used. Then, in Section 3.3.2 we explain how we defined the regions of the input and how we estimate the accuracy of the basic similarity functions, finally in Section 3.3.3 we explain the algorithm addressing the person name disambiguation problem.

3.3.1 Basic similarity functions

Similarity functions associate a value from the interval $[0, 1]$ to a pair of entities. In our case, instead of comparing the entities themselves, we compare the related web-pages. As a preprocessing step we apply information extraction tools, so the input to the similarity functions is the extracted information and not the pages themselves. In other terms, we apply (dictionary-based) named entity recognition techniques.

Each similarity function compares two webpages based on a particular feature (like concepts, URLs etc) using a similarity measure (like cosine similarity, number of overlaps etc) [MRS08], [HFC$^+$08]. We use common observations in coming up with the following similarity functions. Two webpages are about a same person, if the concepts or organizations or person names etc mentioned on the pages are similar/overlap, or if the pages URLs are on a same Web domain.

Regarding the implementation: For extracting features from the webpages we used several information extraction tools, including "alchemy API"[1] to extract named entities, "GATE" [Cun02], "openCalais" [Ope] to extract other types of entities, such as organizations and locations. We also extract wikipedia-based concepts using Textwise[2]. Finally for representing a webpage as document vector we use the services provided by lucene[3]. The similarity functions we consider are summarized in Table 3.3.1.

3.3.2 Quality-aware similarity assessment

3.3.2.1 Accuracy estimations

We estimated the accuracy of individual similarity functions in different ways. These include *global accuracy estimates*, where we give an overall estimate for the entire similarity function and *region-based estimates*, where we partition the input into smaller regions, where we can do estimations much more reliably.

Global accuracy estimation: Given a single similarity function, we can consider two related persons equivalent if their similarity value is higher than an appropriately chosen threshold. Indeed, for each

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

[2] http://www.textwise.com/

[3] http://lucene.apache.org/

Function	Feature	Similarity Measure
F1	Weighted Concept Vector	Cosine Similarity
F2	URL of the page	String Similarity
F3	Most frequent name on the page	String Similarity
F4	Concepts Vector	Number of overlapping concepts
F5	Organizations Entities on the page	Number of overlapping organizations
F6	Other Person-Names on the page	Number of overlapping persons
F7	The name closest to the search keyword	String Similarity
F8	TF-IDF (based weights) words vector	Cosine Similarity
F9	TF-IDF (based weights) words vector	Pearsons Correlation similarity
F10	TF-IDF (based weights) words vector	Extended Jaccard similarity

Table 3.1: Basic similarity function descriptions

function we have chosen such a threshold, and based on the training set, we estimated the accuracy of the threshold-based decision: we computed, what is the percentage of correct matches, if we would consider that two entities with similarity values above the threshold do match.

The accuracy of such decisions clearly depends on the choice of the threshold. For each function, we have chosen a threshold, which –based on the training set– maximizes the number of correct decisions. We used these estimations as a base-line for our experiments.

As we discussed in Section 3.2, as an alternative to global accuracy estimations, we can partition the input to smaller regions and compute accuracy estimates for these parts.

Region-based accuracy estimation: We tried multiple ways to divide the input into regions:

1. We defined the regions based on the similarity values: we divided the similarity values to equal sized sub-intervals: $[0, 0.1), [0.1, 0.2), \ldots, [0.9, 1]$, and one region consists the pairs having the values in a given range. This is a very simple definition, however, the similarity values do not have a uniform distribution in the $[0, 1]$ interval, thus by this definition, some regions contain significantly larger than others.

2. We clustered the similarity values corresponding to the training set using the $k-means$ clustering technique. (We have chosen $k = 15$.) The pairs, whose similarity values fall into one cluster form a region.

In the case of the functions $F5$ and $F6$ we further divided the regions we constructed in this way. The function $F6$ computes the number of overlaps of person names in the corresponding Web documents. If we obtain the value 0, this can have multiple reasons. Either (one of the) Web documents do not contain such person names, or they both contain person names, but the two sets are different. In such cases, we defined the regions using “dimensions”: the similarity value and the existence/non-existence of information.

Based on the training set, for each region we computed an accuracy estimate. From the training sample set, each region would contain certain sample points corresponding to link existence and non-existence. Accuracy for a region is then defined as the percentage of the sample points representing link existence. If this value is lower than 0.5 then it suggests that the majority pairs should not be considered as a link. Note that the accuracy estimations are based on the small training set and not the entire data, so computationally the method remains feasible.

3.3.2.2 Combining multiple functions

Given the heterogeneity of the Web, we cannot expect that we can design a single similarity function which would perform optimally in all cases. To overcome this problem we compute several similarity functions and try to make our decision based on a combination of the similarity functions. To find a suitable way of combination involves a lot of challenges.

The different functions report similarity values with very different value distribution as they capture different aspects of similarity. Thus instead of combining the similarity values themselves, we try to combine the decisions (whether or not to consider two entities as equivalent) and the estimated accuracy values.

In this way, for each function f_i we obtain a graph $G^{f_i}_{D_j}$, together with accuracy estimates, where D_j is the decision criteria, i.e. whether we decide upon a single threshold or also consider the accuracy estimates. Our goal is to combine the the individual graphs $G^{f_i}_{D_j}$ into a single graph $G_{combined}$. First we obtain a multi-graph, where the multiple edges between two nodes correspond to the edges from the individual graphs. We weight the edges with the individual accuracy estimates, which we consider as estimations of the probability of a link. Then we compute a weighted average and obtained an optimal threshold, based on our training set. If the combined value is above this threshold, we add an edge to $G_{combined}$.

We also considered other combination techniques. Instead of considering the weighted average of the values, we used other aggregation functions, namely we have selected the maximum value. Interestingly, this combination technique performed the best on our datasets, which might not always be the case. It is important to note that not always the same function performed the best.

3.3.3 Entity resolution algorithm

We say that two entity references (names) n_i and n_j are equivalent ($n_i \equiv n_j$) if they refer to the same person. Clearly this relation is transitive. The relation of the entity references can be represented as a graph, in which for each entity reference there is a vertex in the graph, and two vertices are connected by an edge whenever the two corresponding entities are equivalent. We refer to this graph as the entity graph. The goal of the entity resolution algorithms is to reconstruct this entity graph as accurately as possible. Note that the entity graph has very specific properties: it is not a connected graph, it is a union of pairwise disjunct connected components and each component is a clique, i.e. a complete graph, because of the transitivity of the equivalence relation.

Our entity resolution technique is the following. First we compute a complete weighted graph $G^{f_i}_w$ for each similarity function f_i. (The nodes of the graph $G^{f_i}_w$ correspond to the Web pages, while the weights on the edges are the similarity values reported by f_i.) To avoid computational bottlenecks, we apply a basic blocking technique, so essentially we only compute the similarity values between documents, which are about a person with the same name.[4] From the graph $G^{f_i}_w$ we would like to obtain a graph G_{D_j}, a (not-weighted) graph, where an edge between two nodes shall indicate whether the entities corresponding to the nodes are the same. This transformation depends on the decision criteria D_j. These decision criteria include to chose values above a threshold or also consider accuracy estimates, as it is explained in Section 3.3.2.1. Once we have all the graphs $G^{i}_{D_j}$, for all functions f_i and all decision

[4] Such blocking strategy is very natural in the datasets we used, where the documents already organized around person names. In general, one needs to consider the applicable blocking schemes more carefully.

criteria D_j, we obtain a combined graph $G_{combined}$, which is explained in Section 3.3.2.2. For this we also use accuracy estimates $acc(G^i_{D_j})$, based on the training set. Finally, we apply clustering techniques to obtain the final entity resolution. In our recent implementation we compute the transitive closure of the graph $G_{combined}$, but we also experimented with several other clustering techniques, such as correlation clustering [BBC04]. The overall procedure is summarized in Algorithm 3.2.

Algorithm 3.2: Quality-aware Entity Resolution Algorithm for Person Name Disambiguation

```
1: compute the graph $G_w^{f_i}$ for each $f_i$ (per block)
2: obtain the decision criteria $D_j$ (threshold, regions, etc.) from the training set
3: apply the decision $D_j$ to the data, to compute $G^i_{D_j}$, for each $i$ and $D_j$
4: compute the accuracy $acc(G^i_{D_j})$
5: combine them, for all $i$, $D_j$
6: apply a clustering algorithm
7: output the final entity resolution
```

3.4 Classifying Twitter messages[5]

In this section we focus on a second problem, where we apply our quality-aware similarity assessment techniques, namely the Twitter classification problem. Twitter [6] is a popular service where users can share short messages (a.k.a. tweets) on any subject. Twitter is currently one of the most popular sites of the Web: as of March 2013, Twitter users send more than 200 million messages per day on average [7]. As users are sharing information on what matters to them, analyzing twitter messages can reveal important social phenomena, indeed a number of recent studies like [GAC$^+$10] report such findings. Clearly, twitter messages are also a rich source for companies, to study the opinions about their products. To perform sentiment analysis or obtain reputation-related information, one needs first to identify the messages which are related to a given company. This is a challenging task on its own as company or product names are often homonyms. This is not accidental, companies deliberately choose such names as part of their branding and marketing strategy. For example, the company Apple Inc. shares its name with the fruit apple, which again could have a number of figurative meanings depending on the context, for example, “knowledge” (Biblical story of Adam, Eve and the serpent) or New York (the Big Apple).

Our task is to relate tweets to a company entity, which can be seen as a special case of the entity matching problem. We assume that we are given a set of companies and for each company a set of tweets, which might or might not be related to the company (i.e. the tweets contain the company name, as a keyword). Constructing such a matcher is a challenging task, as tweet messages are very short (maximum 140 characters), thus they contain very little information, and additionally, tweet messages use a specific language and often also incorrect grammar, they are full with proprietary abbreviations, which are hard to interpret without further background knowledge. To overcome this problem, we constructed profiles for each company, which contain more rich information. For each company, in fact, we constructed

[5] We study Twitter message classification problem in depth in the next chapter (Chapter 4). However, in this chapter, we are presenting this problem in a different context, and try to solve the problem using our proposed framework. For clarity and continuity of the presentation, we repeat essential parts of the problem here.

[6] http://twitter.com

[7] http://articles.washingtonpost.com/2013-03-21/business/37889387_1_tweets-jack-dorsey-twitter

several profiles, some of them automatically, some of them manually. The profiles are essentially sets of keywords, which are related to the company in some way. We also created profiles, which explicitly contains unrelated keywords. Once we have the profiles, we are facing an entity matching problem. In this context, we make use of our quality-aware similarity assessment.

Below, in Section 3.4.1 we give a more precise problem definition. In Section 3.4.2 we explain how we represent Tweet messages and company profiles. We explain in Section 3.4.3 the use of quality-aware similarities and our Twitter classification technique.

3.4.1 Problem Statement

In this section we formulate the problem and our computational framework more formally. The task is concerned to classify a set of Twitter messages $\Gamma = \{T_1, \ldots, T_n\}$, whether they are related to a given company C. We assume that each message $T_i \in \Gamma$ contains the company name as a sub-string. We say that the message T_i is related to the company C, $related(T_i, C)$, if and only if the Twitter message refers to the company. It can be that a message refers both to the company and also to some other meaning of the company name (or to some other company with the same name), but whenever the message T_i refers to company C we try to classify as TRUE otherwise as FALSE. The task has some other inputs, such as the URL of the company $url(C)$, the language of the webpage, as well as the correct classification for a small number of messages (for some of the companies).

For the Twitter classification problem, we assume that we have training sets corresponding for a few companies (C^{TR}). Our goal is to classify test sets corresponding to new (unseen) companies (C^{Test}), for which we do not have training data, i.e. $C^{TR} \bigcap C^{Test} = 0$.

3.4.2 Information representation

The tweet messages and company names alone contain very little information to realize the classification task with good accuracy. To overcome this problem, we created profiles for the companies, several profiles for each company. These set of profiles can be seen as a model for the company. In this section, we discuss how we represent tweet messages and companies and we also discuss how we obtained these profiles. In the classification task we eventually compare a tweet against the profiles representing the company entity.

Tweet Representation

We represented a tweet as a bag of words (unigrams and bigrams). We do not access the tweet messages directly in our classification algorithm, but apply a preprocessing step first, which removes all the stop-words, emoticons, and twitter specific stop-words (such as, for example, RT, @username). We store a stemmed[8] version of keywords (unigrams and bigrams), i.e.

$$T_i = set\{wrd_j\}.$$

[8] Porter stemmer from python based natural language toolkit available at http://www.nltk.org

Company Representation

We represent each company as a collection of profiles, formally

$$E^k = \{P_1^k, P_2^k, \ldots, P_n^k\}.$$

Each profile is a set of weighted keywords i.e. $P_i^k = \{wrd_j : wt_j\}$, with $wt_j \geq 0$ for positive evidence (i.e. keywords, which -if contained in a message- shall indicate that the message is related to the company) and $wt_j < 0$ for negative evidence.

For the tweets classification task, we eventually compare the tweet with the entity (i.e. company) profile. For better classification results, the entity profile should have a good overlap with the tweets. Unfortunately, we do not know the tweet messages in advance, so we tried to create such profiles from alternative sources, independently of the tweet messages. The entity profile should not be too general, because it would result many false positives in the classification and also not too narrow, because then we could miss potential relevant tweets.

We generated most of our profiles automatically, i.e. if one would like to construct a classifier for a previously unseen company, one can automatically generate the profiles. Further, small, manually constructed profiles further improve the accuracy of the classification process.

We used the following profiles: the *homepage profile* contains keywords, extracted from the Web page of the company, the *metadata profile* relies on the metadata of the Web page. The *category profile* contains keywords relevant to the domain of the company. Similarly, for *common-knowledge profile* we obtained relevant keywords from GoogleSets[9]. We also defined *user-feedback-profiles* containing positive and negative keywords from users. For more details on semi-automatic profile construction see Section 4.2.2.2 of Chapter 4. Table 3.2 shows how an “Apple Inc”[10] company entity is represented using different profiles. As we constructed the profile semi-automatically, some of the keywords might be incorrect.

3.4.2.1 Features Extraction

We define a feature extraction function, which compares a tweet T_i to the company entity representation E_k and outputs a vector of features.

$$Fn(T_i, E_k) = \{\ \overbrace{G_1, \ldots, G_m}^{profile-features}\ ,\ \underbrace{F_1, \ldots, F_n}_{tweet-specific}\ , \overbrace{U_1, \ldots, U_z}^{ad-hoc}\}$$

Here the G_i are profile-specific, which are entirely based on the quality of the entity profiles and do not depend on Tweet message T_i. One could use different ways of quantifying the quality of the profiles.

- Boolean: In this work we make use of boolean metrics to represent if a profile is empty or has sufficient keywords.
- Other possibility is that a human can inspect the profiles and assign a metric of x $\in$ [0,1] based on the perceived quality. One could think of exploring an automated way of assigning this number.

[9] GoogleSets `http://labs.google.com/sets` is a service that generates a set of keywords, given a few examples.

[10] http://www.apple.com

Profile Type	Keywords
HomePage Profile	iphone, ipod, mac, safari, ios, iphoto, iwork, leopard, forum, items, employees, itunes, credit, portable, secure, unix, auditing, forums, marketers, browse, dominicana, music, recommend, preview, type, tell, notif, phone, purchase, manuals, updates, fifa, 8GB, 16GB, 32GB,. . .
Metadata Profile	{empty}
Category Profile	opera, code, brainchild, movie, telecom, cruncher, trade, cathode-ray, paper, freight, keyboard, dbm, merchandise, disk, language, microprocessor, move, web, monitor, diskett, show, figure, instrument, board, lade, digit, good, shipment, food, cpu, moving-picture, fluid, consign, contraband, electronic, volume, peripherals, crt, resolve, yield, server, micro, magazine, dreck, byproduct, spiritualist, telecommunications, manage, commodity, flick, vehicle, set, creation, procedure, consequence, second, design, result, mobile, home, processor, spin-off, wander, analog, transmission, cargo, expert, record, database, tube, payload, state, estimate, intersect, internet, print, factory, contrast, outcome, machine, deliver, effect, job, output, release, turnout, convert, river,. . .
GoogleSet Profile	itunes, intel, belkin, 512mb, sony, hp, canon, powerpc, mac, apple, iphone, ati, microsoft, ibm,. . .
UserFeedback Positive Profile	ipad, imac, iphone, ipod, itouch, itv, iad, itunes, keynote, safari, leopard, tiger, iwork, android, droid, phone, app, appstore, mac, macintosh
UserFeedback Negative Profile	fruit, tree, eat, bite, juice, pineapple, strawberry, drink

Table 3.2: Apple Inc Company Profiles

The F_i features are tweet specific features, i.e. they quantify how close a tweet overlaps with the entity profiles. We use a comparison function to compare the tweet message T_i, which is a bag of words, with j^{th} profile P_j^k, which is also a bag of weighted keywords, to get the F_j^{th} feature. In this work we use Boolean overlap as one of the comparison functions, which compares two bags of words looking for exact overlap of keywords, and for all such keywords the sum of their weights quantify how close the tweet message is to the entity profile. Formally with T_i = Set$\{w_1^t, w_2^t, \ldots, w_k^t\}$ and P_j^k = Set$\{w_1^p : wt_1, w_2^p : wt_2, \ldots, w_m^p : wt_m\}$, we compute the F_j feature using the Boolean overlap comparison function as:

$$\begin{aligned} &F_j = BooleanOverlap(T_i, P_j^k) = \sum_q wt_q, \text{ where } q \\ &\text{is the index of overlapping words, i.e.} \\ &w_q^p \in Set\{w_1^t, w_2^t, \ldots, w_k^t\} \bigcap Set\{w_1^p, w_2^p, \ldots, w_m^p\} \end{aligned} \tag{3.1}$$

The above comparison function is simple and easy to realize, but it may miss out some potentially similar words. We also make use of Edit-Distance and Jaro similarity based comparison functions to identify similar words.

The U_i features encapsulate some user based rules, for example, presence of the company URL domain in the tweet URL list, is a big enough evidence to classify the tweet as belonging to the company.

3.4.2.2 Classification Process

The classifier is a function which takes the feature vector as input and classifies the tweet as $\{TRUE, FALSE\}$, with $TRUE$ label if the tweet is related to the company and $FALSE$ otherwise. We use the Naive Bayes Classifier model for designing the individual classifiers. We have chosen to use the Naive Bayes technique, as it was easy to realize and still promises acceptable accuracy. For each company in the training set (C^{TR}), based on the company tweets, we find the conditional distribution of values over features for two classes, for the class of tweets which are related to the company and the another class of tweets, which are not related. With the help of these conditional probabilities, as shown in equations (3.2, 3.3) and by applying Bayes theorem, we can classify an unseen tweet whether it is related to the company or not.

Let us denote the probability distribution of features of the tweets that are related to a given company with

$$P(f_1, f_2, \ldots, f_n \mid C), \tag{3.2}$$

and the probability distribution of features of the tweets that are not related to the company with

$$P(f_1, f_2, \ldots, f_n \mid \overline{C}). \tag{3.3}$$

Then, for an unseen tweet t, using the features extraction function we compute the features values: $(f_1, f_2, \ldots, f_n)$. The posterior probabilities of whether the tweet is related to the company or not, are calculated as in equations (3.4, 3.5).

$$\begin{aligned} P(C \mid t) = \frac{P(C) * P(t \mid C)}{P(t)} = \\ = \frac{P(C) * P(f_1, f_2, \ldots, f_n \mid C)}{P(f_1, f_2, \ldots, f_n)} \end{aligned} \tag{3.4}$$

$$\begin{aligned} P(\overline{C} \mid t) = \frac{P(\overline{C}) * P(t \mid \overline{C})}{P(t)} = \\ = \frac{P(\overline{C}) * P(f_1, f_2, \ldots, f_n \mid \overline{C})}{P(f_1, f_2, \ldots, f_n)} \end{aligned} \tag{3.5}$$

Depending on whether $P(C \mid t)$ is greater than $P(\overline{C} \mid t)$ or not, the naive Bayes classifier decides whether the tweet t is related to the given company or not, respectively.

3.4.3 Entity matching with quality-aware similarities

Given the representation we explained in Section 3.4.2, the Twitter classification can be seen as an entity matching problem. We address the problem with our quality-aware entity matching strategy (Algorithm 3.3). First we design individual classifiers based on our training set. Each of them uses a subset of all features or possible comparison methods. We then identify the regions of the unseen companies and estimate the accuracy of the individual classifiers in these regions. Once we have these accuracy

Set1	Set2	Profiles Used	Comparison Fn
BB1	B1	**X**=[Homepage Profile, Category Profile, Metadata Profile, GoogleSet Profile, UserFeedback Positive Profile, UserFeedback Negative Profile]	BooleanOverlap
BB2	B2	**X**	EditDistance
BB2	B2	**X**	JaroSimilarity
BB4	B4	[Homepage Profile]	**Y**=[BooleanOverlap, EditDistance, JaroSimilarity]
BB5	B5	[UserFeedback Negative Profile]	**Y**

Table 3.3: Individual Classifiers

Algorithm 3.3: Quality-aware Entity Matching Algorithm for Twitter Messages Classification

```
compute decisions using multiple individual classifiers
identify the regions in the feature space for the companies in the test set
estimate the accuracy, for each classifier
combine the decisions of the individual classifiers, using the estimates, for unseen companies
decide whether the entities match
output the decision
```

estimates, we combine the decision of individual classifiers for unseen companies and we use these combined values as a basis for our final decision, whether we consider two entities as a match.

The above algorithm can be seen to proceed in two phases. In the first phase, we construct many individual classifiers, based on the training set. We need many individual classifiers, as we do not know beforehand, which set of features and comparison functions one should use. In the second phase, we chose the best possible classifier for the unseen companies. In this phase, we rely on accuracy estimates of the individual classifier in the “neighborhood” of the unseen companies. We explain these phases for two different scenarios.

1. In the first scenario, the individual classifiers are in the Set1 = $\{$BB1, BB2, BB3, BB4, BB5$\}$. Each classifier in the Set1 is trained *per company and per feature group*. We have in total $|C^{TR}| * 5$ individual classifiers. We consider only the classifiers of the companies that are “similar” to the unseen company. For each unseen company, we consider the $K = 5$ closest companies from the training set as possible candidates, using the dot product distance metric, where each company profile is seen as a vector in the terms-dimension space.

2. For the second scenario, the individual classifiers are in the Set2 = $\{$B1, B2, B3, B4, B5$\}$. We have one classifier per feature set for the entire training set, i.e. in its design it makes use of tweets of all the companies in the training set. In this case, we have 5 individual classifiers in total. We divide the companies in the training set into 6 groups using the k-means clustering technique ($k = 6$). Each cluster is considered as a region and we estimate the accuracy of each classifier for each region. Figure 3.3 depicts the accuracy estimates of the individual classifiers. The figure

suggests that there is no single best classifier and by combining the classifiers we have a chance to achieve better performance. For an unseen company, we first decide to which region it belongs to, by computing its distance to the means of the different regions. As the combination strategy, we choose the most accurate classifier in this region, and we use it as the classifier for the unseen company.

Regarding the computational efforts, in the first case, we are creating many classifiers, but each of them is constructed using a small subset of the training set, containing the relevant company name. In the second case, we have only a few classifiers, each constructed using the entire training set.

Overall, if we have many classifiers making decisions about the entity match, the next question is how can we decide on the final result. One way is to chose the globally most accurate classifier among the many individual classifiers and use it for making the decisions on the test set. The globally most accurate classifier might not necessarily be the best classifier for unseen companies. However, we can do better if we can make use of the accuracy estimates associated with the regions. Other simple alternative combining strategies could be taking the weighted averages, maximal voting, etc. of the individual classifier decisions. For comparison, we also train an SVM Classifier [CST00, MDM07] as a generic classifier, which makes use of all features: profile-features, tweet-specific features and ad-hoc/heuristics-based features, in its classification task.

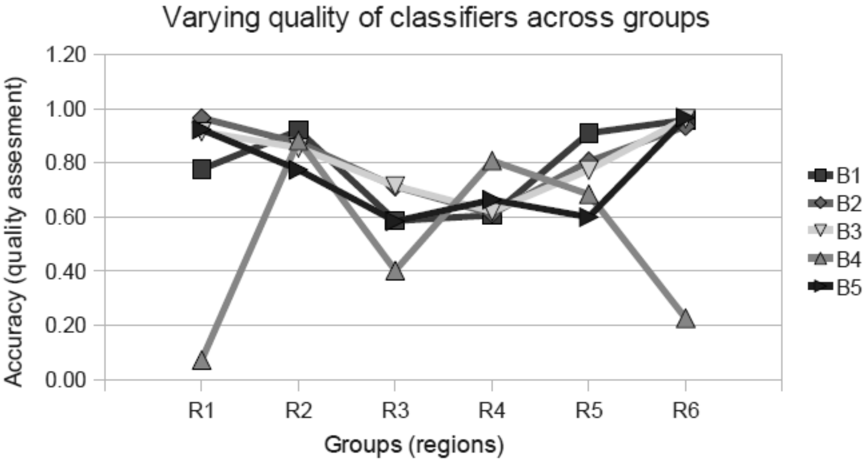


Figure 3.3: Accuracy estimates of the individual classifiers on WePS-3 Twitter Dataset.

3.5 Experimental evaluation

Experimental setup

We performed our experiments on a 2GB RAM, Genuine Intel(R) T2500 @ 2.00 GHz CPU. Linux Kernel 2.6.24, 32-bit machine. We implemented our methods using matlab, java and python.

3.5.1 Person name disambiguation

3.5.1.1 Datasets

For our experiments for evaluating the person name disambiguation we used two different datasets: the WWW'05 people dataset and the WePS people dataset. The WWW'05 dataset was created in [BM05]. This dataset was also used in a series of papers, which enabled us to compare our methods with other techniques. The dataset contains Web documents for 12 different person names. The dataset was created by querying the Web using the google search engine with the different person names. The top 100 returned web documents for the web search were gathered and labeled manually. For each person, the correct resolution is available together with the data. We used this ground truth to measure the quality of our techniques. The number of clusters for each person name is different, it varies from 2 to 61.

WePS people test dataset is provided by the web people search clustering task [WeP09]. The test data consisted 30 Web page collections, each one corresponding to one ambiguous name. These 30 person names were chosen from three different sources: wikipedia, ACL'08 (Association for Computational Linguistics Program committee members) and US census data. Each person name was queried using yahoo search API and the top 150 results were included into the dataset. We have evaluated our techniques on WePS people dataset. We report the performance figures we observed on the 10 person names chosen from the ACL'08.

3.5.1.2 Measures of interest

Various measures are considered to assess the quality of entity resolution. Precision, recall and F-measure are widely used in information retrieval. We also measure the Rand-index [MRS08] and the F_p-measure [HFC$^+$08], which is the harmonic mean of purity and inverse purity. They are typically measures from information retrieval or variants of those measures. We summarize here the definitions. Some of these definitions can be found in [MRS08].

An entity resolution algorithm tries to predict the entity graph. Given a prediction graph, one can categorize its links with respect to the ground truth, i.e. the correct entity graph, into four categories: true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). The true positives are links which are correctly predicted while the wrongly predicted links are the false positives. Similarly, the correctly predicted missing links fall into the true negatives category, while wrongly predicted missing links are false negatives. We also denote the number of links in the corresponding category with TP, TN, FP, FN.

Precision (P), recall (R) and F-measure (F) are defined as:

$$P = \frac{TP}{TP + FP},$$

$$R = \frac{TP}{TP + FN},$$

$$\text{and } F = \frac{2PR}{P + R}$$

Accuracy (a.k.a. Rand index, RI) is the percentage of correct decisions for the predicted links:

$$RI = \frac{TP + TN}{TP + TN + FP + FN}$$

F_p-measure (F_p) is the harmonic mean of purity and inverse purity. Purity is defined as follows [HFC$^+$08]: let $M = \{M_1, \ldots, M_n\}$ be the clusters of the ground truth and let $C = \{C_1, \ldots, C_m\}$ be the clusters predicted by the algorithm and let $Prec(C_i, M_j)$ denote the precision of C_i w.r.t. M_j. Purity is defined as

$$Pur(C, M) = \sum_{C_i \in C} \frac{\mid C_i \mid}{\mid C \mid} \max_{M_i \in M} Prec(C_i, M_j).$$

while inverse purity [11] as

$$IPur(C, M) = \sum_{M_i \in M} \frac{\mid M_i \mid}{\mid M \mid} \max_{C_i \in C} Prec(M_i, C_j).$$

We note here that the above measures rely on the fact that we know the ground truth, which is unrealistic in the Web context. We could apply them for the document collections in our experiments, as we had this information available.

3.5.1.3 Methods

Given the dataset, we use 10% of the complete dataset as the training set. The performance of the entity resolution (entity matching) algorithm depends on how well the training set represents the features of the complete dataset. In order to avoid any bias, we repeated the experiments for 5 runs and the averages of the observed results are presented. On each run we randomly choose the training subset from the complete dataset. We make use of a standard 10-fold-cross validation technique to obtain the optimal parameters of a classifier. For computing the parameters we minimize the loss function, i.e. the number of incorrect decisions.

3.5.1.4 Experimental results

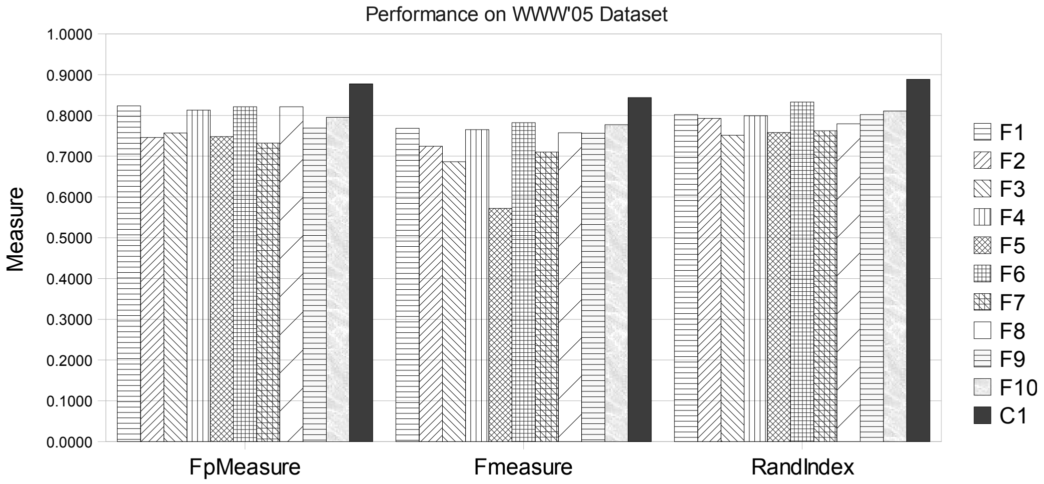


Figure 3.4: WWW'05 people dataset results graph.

[11] The name "inverse purity" is supported by the fact that $Pur(C, M) = IPur(M, C)$.

Figure 3.4 shows the performance of the individual similarity functions on the entire WWW'05 dataset. The figure shows three metrics, namely F_p-measure, F-measure and $Rand$-index. The final column, depicted as black in the figure, is the combined performance of our quality-aware combination technique, which clearly shows improved performance. Similarly, Figure 3.5 shows the experimental results on the WePS peoples dataset.

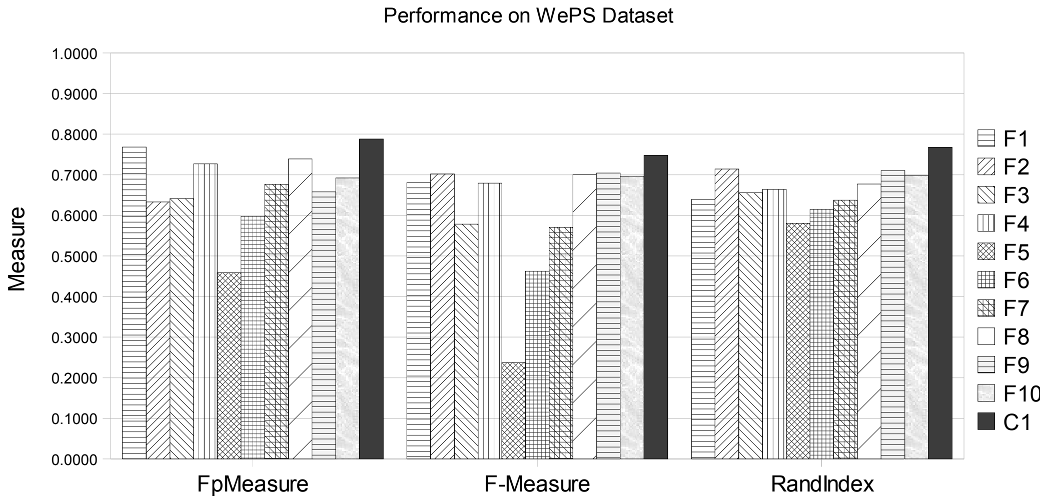


Figure 3.5: WePS people dataset results graph.

Table 3.4 contains the achieved F_p values, for each individual person, by each individual function in the WWW'05 dataset. One can observe that each function performs differently for different persons. For example, for "Voss" the function F8 has the highest F_p-value, while for "Mulford" the best function is F6.

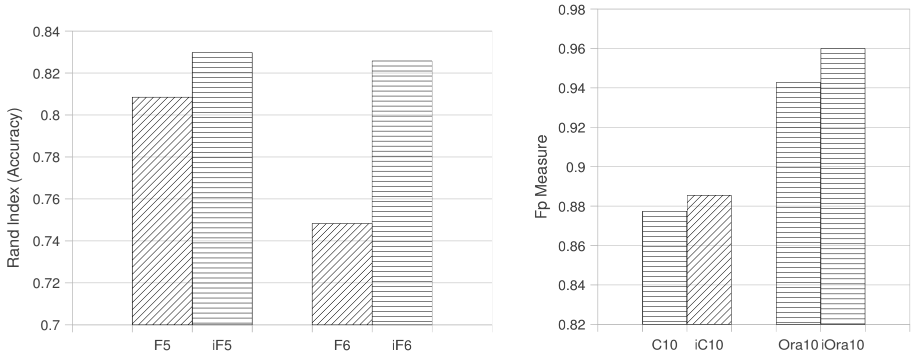
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	C10	W
Cheyer	0.9686	0.9948	1.0000	0.9686	0.7950	0.9948	1.0000	0.9948	0.9948	0.9948	1.0000	0.9948
Cohen	0.8724	0.3827	0.7368	0.8859	0.8444	0.8991	0.8839	0.8746	0.8746	0.8718	0.8991	0.8816
Hardt	0.8680	0.8828	0.8985	0.8680	0.4717	0.9074	0.8985	0.8828	0.8828	0.8779	0.9074	0.8828
Israel	0.8206	0.7568	0.7881	0.8312	0.8093	0.8476	0.7257	0.8315	0.7536	0.7568	0.8476	0.8690
Kaelbling	0.9831	0.9944	0.9711	0.9831	0.9012	0.9467	0.9711	0.9944	0.9888	0.9944	0.9944	0.9944
Mark	0.7871	0.7871	0.7228	0.7871	0.7871	0.7871	0.7668	0.7871	0.7915	0.7871	0.8104	0.7871
Mccallum	0.7921	0.7391	0.6642	0.7812	0.8066	0.8248	0.4667	0.8024	0.5851	0.8187	0.9670	0.8597
Mitchell	0.8473	0.7756	0.5796	0.7417	0.7981	0.7733	0.4448	0.5966	0.7097	0.7382	0.8575	0.6448
Mulford	0.7471	0.7467	0.7569	0.7471	0.7337	0.7582	0.7569	0.7467	0.7467	0.7467	0.8053	0.7467
Ng	0.8607	0.7111	0.7493	0.8660	0.7938	0.8163	0.7031	0.8086	0.7082	0.7082	0.8813	0.8845
Pereira	0.7215	0.5420	0.6362	0.7180	0.6389	0.6942	0.5571	0.7326	0.5554	0.5519	0.7573	0.7438
Voss	0.6094	0.6365	0.5813	0.5760	0.5993	0.6073	0.6135	0.8016	0.6391	0.6979	0.8016	0.7567

Table 3.4: F_p measure for each name in WWW'05 Dataset

Dataset	Metric	I4	I7	I10	C4	C7	C10	W	Related work
WWW'05	F_p-measure	0.8128	0.8211	0.8232	0.8537	0.8732	0.8774	0.8371	0.864 [KCMN08], 0.9000 [CKM09]
	F-measure	0.7654	0.7773	0.7822	0.8338	0.8376	0.8438	0.8168	0.8000 [BM05], 0.8 [CKM09]
	RandIndex	0.8018	0.8109	0.8326	0.8747	0.8814	0.8886	0.8531	
WePS	F_p-measure	0.7270	0.7388	0.7682	0.7560	0.7659	0.7880	0.7785	0.791 [KCMN08], WePS: 0.7800
	F-measure	0.7042	0.7042	0.7042	0.7127	0.7231	0.7476	0.7190	
	RandIndex	0.7102	0.7102	0.7139	0.7492	0.7531	0.7675	0.7290	

Table 3.5: Comparison of results of the ensemble classifiers w.r.t. the state-of-the-art classifiers for WWW'05 and WePS people datasets.

Table 3.5 shows that by considering more and more functions we indeed get a better performance for both datasets. The first three columns show the maximal performance considering just the threshold-based technique, by including functions $I4$=$\{F4$,$F5$,$F7$,$F9\}$, $I7$=$\{F3$, $F4$, $F5$, $F7$, $F8$, $F9$, $F10\}$, and $I10$=$\{F1$,...,$F10\}$, respectively. The columns $C4$, $C7$ and $C10$ take the same functions as the first three columns respectively, but there we chose the best decision criteria, based on accuracy estimation of the regions. The column W shows the performance of weighted average combination result. The table also contains a comparison with the figures reported in the literature. The best results for the WWW'05 dataset were reported in the paper [CKM09], however they manually improved the available ground truth (and the improved data is not public), therefore the comparison is not precise. The last column contains the result achieved by the WePS competition winner. We found this result in [KCMN08], but we could not obtain the original reference.



(a) Rand Index

(b) F_p-measure

Figure 3.6: (a) Improvements in the Rand index for individual functions. (Basic vs. improved regions.) for WWW'05 peoples dataset; and (b) F_p-measure of the combined classifier and the maximum plausible performance. (Basic vs. improved regions.) for WWW'05 peoples dataset.

Figure 3.6(a) depicts the Rand index values, for two similarity functions $F5$ and $F6$ (Person overlap, Organization overlap, Table 3.3.1), on the WWW'05 dataset. For both functions, the left bar (F5, F6) is the mean Rand index value across all the person names, where we were relying on regions defined by similarity values only (basic regions). For obtaining the value depicted in the right bar (iF5,iF6), we refined the regions and also consider whether a value is missing or not (improved regions). The improved performance is due to the refinements in the definition of the regions. Low similarity values can have many reasons; here we distinguish the cases where information is missing from the cases, where the information is dissimilar. Considering this feature does not increase the number of regions, thus it does not increase the computational efforts needed for computing the accuracy estimates.

We wanted to see, how far are our methods from the accuracy values that we could have potentially achieved using the same similarity functions. We defined an oracle combination function as follows. Whenever one of the classifiers has a decision agreeing with the ground truth, the oracle decides on that value. We computed the accuracy of this oracle combination function (Ora10, iOra10) and we depict it together with C10 and iC10 in Figure 3.6(b).

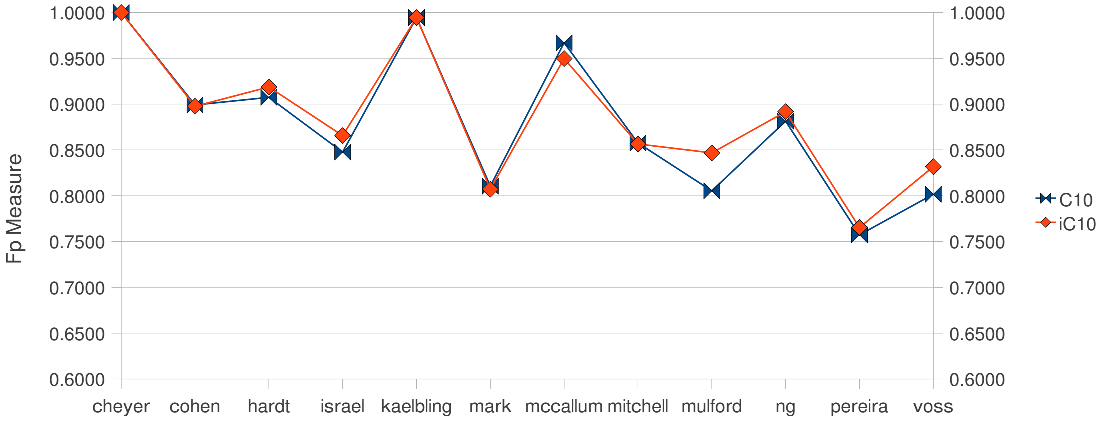


Figure 3.7: Improvement of combined values (Basic vs. improved regions) for WWW'05 peoples dataset.

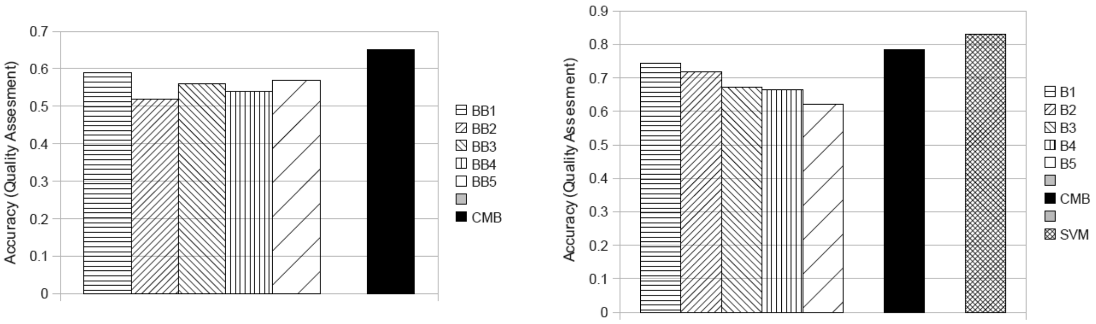
Figure 3.7 depicts the improvements in the combination. The value iC10 is obtained using the refined regions, $\{F1, \ldots, iF5, iF6, \ldots, F10\}$ i.e. as opposed to $I10$, the functions $F5$ and $F6$ are replaces by $iF5$ and $iF6$.

3.5.2 Twitter classification

3.5.2.1 Dataset

Our experimental setup was the following. For our experiments we used the WePS-3 Twitter dataset, which is is available here[12]. We are given a general training set, which consists tweets related to about 50 companies (we denote this set as C^{TR}). For each company $c \in C^{TR}$ we are provided around 400 tweets with their corresponding ground truth, i.e. if the tweet is related to the company or not. For each company, we are provided with the following meta-information: URL, Language, Category. We have trained a generic *classifier* based on this training set. The test set for this task consisted tweets of around 50 new companies. We denote this set of companies as C^{Test}. There was no overlap with the training set,

[12] http://nlp.uned.es/weps/weps-3/data



(a) Individual classifiers vs. quality-aware combination (b) Individual feature set classifiers vs. quality-aware combination

Figure 3.8: (a) Individual classifiers vs. quality-aware combination for WePS-3 Twitter dataset; and (b) Individual feature set classifiers vs. quality-aware combination. for WePS-3 Twitter dataset.

$C^{TR} \bigcap C^{Test} = 0$. For each company $c \in C^{Test}$ there are about 400 Tweets, which are to be classified. We classified them with the classifiers explained in Section 3.4.2.2.

3.5.2.2 Experimental results

The task is of classifying the tweets into two classes: one class which represents the tweets related to the company (positive class) and second class represents tweets that are not related to the company (negative class). Our performance metric for the evaluation was accuracy.

We conduct two series of experiments. In the first case we design five classifiers (BB1, …, BB5) per company in the training set. For each training set company, we compute how a particular classifier performs in its neighborhood and take it as an accuracy estimate for that classifier in that region. Given these set of individual classifiers, we would like to see the performance of quality-aware combination against the case of choosing the most accurate classifier for the complete test set. The results are shown in Figure 3.8(a).

In the second series of experiments, we design five global classifiers (B1, …, B5) for the complete training set. These five global classifiers would have different performance on each individual company in the training set. This performance is taken as an accuracy estimate of a particular global classifier for that region. For quality-aware combination, we choose the global classifier with most accuracy in the region of the company in consideration. We also show the SVM classifier, which makes the quality-aware combination decision implicitly, as a comparison. The results are depicted in Figure 3.8(b). We compare these quality-aware combination against these five global classifiers used against all the test set.

In both figures, we see that quality-aware combination techniques outperform the other techniques which do not take regions based accuracy into consideration.

The amount of accuracy one could achieve with our technique depends on two factors. They are the quality of similarity functions and the type of combination function. We show that starting with a rich set of similarity functions and even with a choice of a simple combination technique we can get results better the state-of-art-techniques. In some cases, more sophisticated combination techniques, like SVM, achieve further improvements.

3.6 Related work

Our work addresses entity resolution problem in Web context. In this section we relate our method with other entity resolution and Twitter classification techniques. We also relate our work to the literature on combining multiple classifiers.

3.6.1 Entity matching

The entity resolution and related problems, such as for example duplicate detection have an extensive literature in the database community, a few important references include [FS69], [HS95], [VME03], [DHM05], [IVE07], and [KR10]. We provided extensive overview of these approaches in Section 2.2 of Chapter 2. Many papers suggest (for example [HS95]) incremental clustering-based methods, while others propose pairwise comparison-based techniques. A recent paper [MBGM06] presents a pairwise comparison-based method, where the authors also consider confidence values during the resolution process. They propose to merge database records, which refer to the same entity, right away, as they are found to be equivalent by the algorithm. The algorithm also computes a new combined confidence value for the merged record. A more complete analysis of results can be found in [BGMM$^+$09], where the authors also study, how to chose the sequence of the records to be processed, such that the running time of the algorithm remains low. In our work, we do not merge or recompute the similarity values.

Chauduri et al. [CGM05] introduce a model for detecting fuzzy duplicates in databases. They extended their model also to a more general setting in [CSGK07]. Their paper is particularly important from methodological point of view, as they systematically derive their entity resolution algorithms from an axiomatic model. Unfortunately their model cannot be easily extended to the Web context because the properties of similarity functions for entities in Web documents do not show the same properties as in the case of fuzzy duplicates, so the basic assumptions of their model are not satisfied.

3.6.2 Entity matching on the Web

Entity resolution in Web context was studied by Kalashnikov et al. [KM06]. They propose to create an entity resolution graph, using the feature-based similarities. The graph witnesses the uncertainty of the features by having multiple nodes, the so called "choice nodes" are corresponding to possible references to a given entity. The authors apply heuristic graph measures to measure the connectedness of entities. The underlying idea behind their heuristic is the "context attraction principle": if two entities are related, then it is likely that there are multiple chains in the entity resolution graph between their corresponding nodes. The authors further improved their techniques in [KCMN08]. In [KCMN08] and in many other approaches, such as for example in [DHM05], the authors consider a more complex graph, which captures more complex relations, rather than the similarities between the entities as in our work. We limited ourselves to a simple representation and to focus the issues in this simpler case, our framework could be later extended to a more complex setting. Their work and their use of context information in [KM06] is a similar technique to our quality-aware similarity assessment technique. We rely on different features, which are also easier to estimate.

Cudré-Mauroux et al. [CMHJ$^+$09] take a different approach to entity resolution in the Web context. They propose a graphical model-based probabilistic framework to capture the relations among the entities. Their framework also includes trust assessments about the providers of the entity equivalence assertions. These trust assessment values are later adjusted as their probabilistic reasoning framework eliminates the detected inconsistencies. While this approach has many advantages, it is not fully applicable to our case, as the underlying factor graph model would have very large cliques, as subgraphs, which could easily lead to poor convergence of the probabilistic reasoning.

On the Semantic Web person names might be annotated with a globally accepted ontology. This direct link between the ontology helps to disambiguate the person names. However, such globally accepted ontologies are not present in the emerging Semantic Web. Instead, ontologies are very often used as local schemas, thus one needs to relate the existing annotation to the ontology one would like to use. The Semantic Web community has developed a plethora of such techniques, see [ES07]. The OKKAM project suggests a different approach, [BPSV09]. They propose a service, which provides globally unique identifiers on large scale for entities, for (semantic) web applications. Their approach relies on the existence of a large and clean (i.e. resolved) collection of entity profiles. Entity profiles collect relevant attributes of real world entities. Our techniques can contribute to create or extend such an entity profile collection.

Balog et al. in [BAdR08] address the problem of clustering web documents based on the person entity, typically needed for web people search task. They explore and empirically evaluate different clustering techniques. In our approach we propose techniques to improve the overall accuracy of the clustering methods.

3.6.3 Combining classifiers

Combining multiple classifiers is studied in the machine learning and also in data mining community [SE10]. The techniques can be broadly divided into two main categories:

1. Classifiers fusion, in which the final decision on a sample point is based on the fusion of decisions of individual classifiers, in some sense similar to achieving consensus. Examples include majority voting, weighted voting.

2. Dynamic Classifier selection: In this scenario, the decision of one of the classifier is chosen as the combined decision. Here, the classifier is chosen based on which classifier best represents the sample point.

Our methods use both combination techniques.

There are different ways have been proposed how to identify regions for accuracy estimates. Woods et al. [WKB97] discuss a method, which divides the sample space into partitions either on predefined criteria or on the features. Each classifiers performance is estimated for each partition. This estimates would be used in choosing a best classifier for each partition. Liu et al. [LY01] propose a novel way of combining classifiers: which is by a technique called as clustering and selection. The input sample space is partitioned into several regions and clustering the correct and incorrect decisions separately. Each classifier performance is estimated for each region. On seeing a new sample, the region to which it belongs to is identified and the classifier with best performance for that region is chosen for the final decision. Strehl et al. [SGC02] address the problem of combining multiple partitions of a set of objects into a single consolidated clustering without accessing the features or algorithms that created the partitions. In our work we use conceptually similar techniques as the papers above, but the actual definitions are specific to the application.

Chen et al. [CKM09] studied the combination of multiple classifiers, where the classifiers are applied for performing entity resolution. They also suggest that the performance of the classifiers depends on the context. Their method introduces techniques to exploit the context and find regions, where the classifier work better. Their method highly depends on their estimation of the total number of clusters (entities), which can be highly unreliable. Once they obtained the combination of the clustering methods, they also apply further techniques to improve their method, such as correlation clustering [BBC04] and related heuristic approximation techniques. Their overall strategy is similar to ours, but their way of defining regions and combining similarity values is different.

Bilenko et al. [BM03] propose to use SVM classifiers for entity matching in databases. They also adapt the distance functions, which are string similarity functions, with the help of machine learning techniques. A multiple classifier approach was used by Zhao et al. [ZR05] for entity identification in heterogeneous database integration scenarios. They also consider various classifier combination techniques to improve the classification accuracy. Our work applies similar techniques, but in a more general context.

Bi et al. [BGB08] propose a classifier combination technique, based on Dempster-Shafer theory of evidence and evidential reasoning. We did not consider this approach, as our classifiers are often not independent.

3.6.4 Twitter classification

The classification of tweets has already been addressed in the literature, in different contexts. Some of the relevant works include [SFD$^+$10], [SST$^+$09], [PP10], [JZSC09].

In [SFD$^+$10], the authors take up the task of classifying the tweets from twitter into predefined set of generic categories such as News, Events, Opinions, Deals and Private Messages. They propose to use a small set of domain-specific features extracted from the tweets and the user's profile. The features of each category are learned from the training set. This task which can be seen as a supervised learning scenario is different from our current task which is a generic learning task.

The authors in [SST$^+$09], build a news processing system based on Twitter. From the twitter stream they build a system that identifies the messages corresponding to late breaking news. Some of the issues they deal with are separating the noise from valid tweets, forming tweet clusters of interest, and identifying the relevant locations associated with the tweets. All these tasks are done in an online manner. They build a naive Bayes classifier for distinguishing relevant news tweets from irrelevant ones. They construct the classifier from a training set. They represent intermediate clusters as a feature vector, and they associate an incoming tweet with cluster if the distance metric to a cluster is less than a given threshold.

In [JZSC09] and [PP10], the authors make use of twitter for the task of sentiment analysis. They build a sentiment classifier, based on a tweet corpus. Their classifier is able to classify tweets as positive, negative, or neutral sentiments. The papers identify relevant features (presence of emoticons, n-grams), and train the classifier on an annotated training set. Their work is complementary to ours: the techniques proposed in our work could serve as an essential preprocessing step to these sentiment or opinion analysis, which identifies the relevant tweets for the sentiment analysis.

The work in this chapter partially relies on the entity profiles and datasets used for Twitter classification task, which we cover extensively in Chapter 4. However, we realized different experiments and we used different classifiers. In this work we use our quality-aware similarity assessment technique and define regions to improve the quality of our accuracy estimations, which was not studied before. The focus of the next chapter is the dynamic maintenance and improvement of company profiles, that is not used in this work.

3.7 Conclusion and future work

We studied two variants of the general entity resolution problem in the Web context, namely the person name disambiguation and the Twitter classification problem. Such entity resolution tasks are essential for realizing the entity-oriented view of Semantic Web. In order to process the information on the Web automatically, one needs to connect the entities present in unstructured Web documents to descriptions of entities, or to entity collections. We designed a simple algorithmic framework for both problems.

We studied the design of similarity assessment techniques. Our proposed method estimates the quality of available similarity values, for particular regions of the input and not globally, as the assessment techniques themselves produce results of different quality. Also it takes specifically into account if some information is not available or missing, which is very common in the context of Web documents. We demonstrated the effectiveness of these methods in our framework: for both problems our techniques show promising results. Quality-aware similarity functions can be used in combination with other algorithmic frameworks as well. The systematic quality assessment and quality-aware combination technique results improved similarity values and improves the overall performance of these algorithms. Clearly, there is a balance between the definition of regions and computational efficiency. Our way of defining regions is simple and easy to realize, yet different from other techniques. Through these techniques we addressed entity resolution for web documents, one of the important problems in realizing entity oriented view of the Semantic Web, in this chapter.

In our future work, we would like to find other ways for defining regions for accuracy estimations. We also plan to address the effect of incomplete information available in the Web pages on the accuracy of the similarity functions even more directly, by considering entropy based metrics, similar to [CMBHA08]. We also would like to extend our quality estimations to more dynamic settings, which is essential if the Twitter messages have to be classified on the fly, as they arrive in the Twitter stream.

Part III

Entities in Microblogging Posts

Chapter 4

Entity-based Classification of Twitter Messages

There's more value in messages shared publicly because more opportunities arise. A kind of social alchemy takes place when a seemingly valueless message finds its way to someone for whom it strikes a chord.

@toomuchnick

Twitter is a popular micro-blogging service on the Web, where people can enter short messages, which then become visible to some other users of the service. While the topics of these messages varies, there are a lot of messages where the users express their opinions about some companies or their products. These messages are a rich source of information for companies for sentiment analysis or opinion mining. There is however a great obstacle for analyzing the messages directly: as the company names are often ambiguous (e.g. apple, the fruit vs. Apple Inc.), one needs first to identify, which messages are related to the company. In this chapter, we address the problem of Entity Matching in Twitter streams. We are interested in deciding if a tweet containing an entity mention is related to a particular real world entity. As seen in the previous chapter, the algorithms for identifying entities in an unstructured text, rely on exploiting the context surrounding the entity-mention for reliably identifying the entities. Twitter messages (tweets) being short messages either have very little context or no context, provide an additional challenge for the entity identification algorithms.

We present various techniques for classifying tweet messages containing a given keyword, whether they are related to a particular company with that name or not. We first present simple techniques, which make use of company profiles, which we created semi-automatically from external Web sources. Our advanced techniques take ambiguity estimations into account and also automatically extend the company profiles from the twitter stream itself. We demonstrate the effectiveness of our methods through an extensive set of experiments. Moreover, we extensively analyze the sources of errors in the classification. The analysis not only brings further improvement, but also enables to use the human input more efficiently.

4.1 Introduction

Twitter[1] is a popular micro-blogging service on the Web, where people can enter short messages (a.k.a. tweets), which then become visible to other users. Twitter is currently one of the most popular sites of the Web: as of March 2013, Twitter users send more than 200 million messages per day on average [2]. While the subject of these varies, in many cases the messages express opinions about companies or their products. Since the service is very popular, the twitter messages form a rich source of information for companies about how their customers like their products. In the same way companies might learn what is the general perception of the company. There is however a great obstacle for analyzing the data directly: as the company names are often ambiguous, one needs first to identify, which messages are related to the company. This name ambiguity is not accidental, the choice of the company name is part of the branding and marketing strategy. Examples for such company and brand names from the technology industry are Apple ™ Inc., Orange® or BlackBerry®.

Hash-tags are often used in twitter messages, as an indirect way of linking tweet messages that are about a common thing (*an event, a news article, an entity, a product,* etc.). It is possible to associate an hast-tag corresponding to an entity, for example: *#apple* hash-tag could represent Apple Inc. company entity. If all tweet messages related to Apple company entity use *#apple* hash-tag then it is trivial to identify all such messages. But in reality, only small set of tweets contain such hash tags, due to which one fails to find all the relevant tweets corresponding to an entity. In this chapter, we are interested in identifying if a tweet – which may or may not contain any hash-tag – is relevant to a particular company entity.

In this work we focus on the problem of classifying twitter messages containing a given keyword, whether or not they are related to a given company. Constructing such a classifier is a challenging task, as tweet messages are very short (maximum 140 characters), thus they contain very little information, and additionally, tweet messages use a specific language, often with incorrect grammar and specific abbreviations, which are hard to interpret by a computer. To overcome this problem, we constructed profiles for each company, which contain more rich information. For each company we collected keywords from different sources (Web, User) automatically and in some cases manually. The company profiles essentially contain these keywords, which are related to the company in some way. With each profile we also maintain a set that contains unrelated keywords. With the help of these profiles we could construct a classifier.

Tweet-ID	**Tweet Message**	**Classification(T/F)**
T1	".. **installed** yesterdays **update released** by *apple*.."	T
T2	".. the *apple* **juice** was bitter.."	F
T3	".. it was easy when *apples* and **blackberries** were only **fruits**.."	T
T4	".. dropped my *apple*, mind u its not the **fruit**.."	T

Table 4.1: Tweets containing the keyword "apple"

Table 4.1 gives some examples of tweets containing the keyword "apple". Our task is to decide whether these messages are related to the company Apple Inc. or not. This task is not trivial, even for

[1] `http://www.twitter.com`

[2] http://articles.washingtonpost.com/2013-03-21/business/37889387_1_tweets-jack-dorsey-twitter

human inspectors. The human decision process relies on some specific keywords, which –together with the background knowledge– give hints for the decision. In the table, the bold words are examples for such possible hints. In our classification techniques, we try to construct profiles, which contain exactly these keywords. Note that in the sentences T3 and T4 the speaker exploits the multiple possible interpretations of the word “apple”. (If one of them is the company Apple Inc. we try to classify the message as TRUE.)

Beyond this standard technique we construct more sophisticated classifiers as well. First we estimate the overall ambiguity of a company name, and include this information in our classification decision. Moreover we do not use static profiles for the companies, rather dynamic ones, which we continually update from the twitter stream. This extension is essential and specific to our classification problem. The keywords appearing in the tweets are repeated with changing frequencies: for example if a company launches a new product, this new product name might appear more frequently in the twitter stream, and such keywords can be temporarily good indications that the message is related to the company. We conducted an extensive set of experiments using the WePS-3 dataset[3] and also through direct access to the twitter stream. The experiments show promising performance figures. Moreover, we extensively analyze the sources of errors in the classification. The analysis not only brings further improvement, but also enables to use the human input more efficiently.

The rest of the chapter is organized as follows. Section 4.2 explains the problem more formally. Section 4.3 presents our basic classification technique, while Section 4.4 describes our more advanced techniques, where we involve ambiguity estimations and also active profiles. Section 4.5 contains the results of our extensive experimental evaluation. Section 4.6 elaborates on the reasons of errors in the classification and presents systematic techniques to minimize the effect of certain types of errors. Section 4.7 summarizes the related work and finally, Section 4.9 concludes the chapter.

4.2 Model and Problem Statement

4.2.1 Problem statement

In this section we formulate the problem and our computational framework more formally. The task is concerned to classify a set of Twitter messages $\Gamma = \{T_1, \ldots, T_n\}$, whether they are related to a given company C. We assume that each message $T_i \in \Gamma$ contains the company name as a sub-string. We say that the message T_i is related to the company C, $related(T_i, C)$, if and only if the Twitter message refers to the company. We also use the term that a tweet belongs to a company, by which we mean the same. It can be that a message refers both to the company and also to some other meaning of the company name (or to some other company with the same name), but whenever the message T_i refers to company C we try to classify as TRUE otherwise as FALSE. We assume that some basic further information is available as input, such as the URL of the company $url(C)$, the language of the Web page.

4.2.2 Model

4.2.2.1 Tweet Representation

We represent a tweet as a bag of words (unigrams and bigrams). We do not access the tweet messages directly in our classification algorithm, but apply a preprocessing step first, which removes all the stop-

[3] `http://nlp.uned.es/weps/weps-3` In fact, we are not using the training set of WePS-3, just the test set with the available ground truth, for evaluation purpose.

words, emoticons, and twitter specific stop-words (such as, for example, RT,@username). We store a stemmed[4] version of keywords (unigrams and bigrams). Formally we have:

$$T_i = set\{wrd_j\}. \tag{4.1}$$

4.2.2.2 Company Representation

We represent each company entity as a profile, where a profile is a set of weighted keywords.

$$P_c = \{wrd_j : wt_j\} \tag{4.2}$$

with $wt_j \geq 0$ for positive evidence keywords (i.e. those words which suggest that the message should be related to the company) and $wt_j < 0$ for negative evidence keywords. We can consider the profile as two sets of weighted keywords. The set with positive weights constitute positive evidence keywords and the set with negative weights represent negative evidence keywords.

$$P_c.Set^+ = \{wrd_j : wt_j \mid wt_j \geq 0\} \tag{4.3}$$

$$P_c.Set^- = \{wrd_j : wt_j \mid wt_j < 0\} \tag{4.4}$$

The weights wt_j corresponding to word wrd_j essentially captures the conditional probability of the event that a message containing the keyword belongs (or does not belong) to the given company C. (For simplicity, we denote these events as C and $\overline{C}$).

$$P(wrd_j \mid C) = wt_j \text{ if } wt_j \geq 0, \tag{4.5}$$

$$P(wrd_j \mid \overline{C}) = |wt_j| \text{ if } wt_j < 0, \tag{4.6}$$

4.2.2.3 Classification Process

For the tweets classification task, we compare the tweet with the entity (i.e. company) profile. We make use of Naive Bayes Classifier [Hec96], [Lew98] for our classification process. We assume the words appearing in a tweet independently contribute towards the evidence of whether the tweet belongs to the company, or not.

For each tweet T_i = $set\{wrd^i_j\}$ we compute the conditional probabilities $P(C \mid T_i)$ and $P(\overline{C} \mid T_i)$ for deciding if a tweet belongs to a company C or not. We make use of Bayes theorem for computing these terms.

$$\begin{aligned} P(C \mid T_i) &= \frac{P(C) * P(T_i \mid C)}{P(T_i)} \\ &= \frac{P(C) * P(wrd^i_1, \ldots, wrd^i_n \mid C)}{P(T_i)} \\ &= K_1 \prod_{j=1}^{n} P(wrd^i_j \mid C) \end{aligned} \tag{4.7}$$

[4] We used the Porter stemmer from the python based natural language toolkit, available at `http://www.nltk.org`

Similarly we have,

$$P(\overline{C} \mid T_i) = K_2 \prod_{j=1}^{n} P(wrd_j^i \mid \overline{C}) \tag{4.8}$$

where, $P(wrd_j \mid C)$ and $P(wrd_j \mid \overline{C})$ are the weights associated with the words wrd_j as described in previous section. Depending on whether $P(C \mid T_i)$ is greater than $P(\overline{C} \mid T_i)$ or not, the Naive Bayes Classifier decides whether the tweet T_i is related to the given company or not, respectively.

4.3 Basic Twitter classification

In this section we present a basic classification technique for twitter messages. This technique is an improved version of our classifier [YMA10a], which we developed in the context of WePS-3 evaluation challenge. It is referenced with the name LSIR-EPFL in [AAG$^+$10]. Our classifier is essentially a Naive Bayes classifier, which relies on constructed company profiles. In the following we give details about how we constructed the profiles from different information sources. We represent a company using basic profile, which is set of weighted keywords. We assume that for each company we are provided with the company name, an URL representing the company, the category to which the company belongs. For each information source we show how we extract the keywords, and discuss the advantages and disadvantages associated with that source.

Homepage Keywords For each company name, we assume that the company homepage URL is available. To extract relevant keywords from the homepage URL, we crawled all the relevant links up to a depth of level (d=2), starting from the given homepage URL. First we extracted all the keywords present on these relevant pages, then we removed all the stop-words, finally we store in the profile the stemmed version of these keywords. From this construction process one would expect that homepage provides us all the important keywords related to the company. However, since the construction is an automated process, it was not always possible to capture good quality representation of the company for various reasons like: the company webpages may use java-scripts, some use flash, some company pages contain irrelevant links, most of the webpages are non-standard home-pages etc. The collected keywords from this source contribute towards positive evidence.

Metadata Keywords HTML standards provides few meta tags[5], which enables a Web page to list set of keywords that one could associate with the Web page. We collect all such meta keywords whenever they are present. If these meta-keywords are present in the HTML code, they have high quality, the meta-keywords are highly relevant for the company. On the negative side, only a fraction of webpages have this information available. The metadata keywords contribute towards positive evidence.

Category Keywords The category, to which the company belongs, is a good source of relevant information of the company entity. The general terms associated with the category would be a rich representation of the entity. For example Apple Inc. belongs to "Computers Software and Hardware" category. One usually fails to find this kind of category related keywords on the homepage

[5] $http://www.w3schools.com/html/html\_meta.asp$

URLs. Further, we make use of WordNet[6], a network of words, to find all the terms linked to the category keywords. Thus by using this kind of source helps us associate keywords like: software,install, update, virus, version, hardware, program, bugs etc to a software company entity. This source of keywords contribute towards positive evidence.

GoogleSet/CommonKnowledge Keywords GoogleSet is a good source of obtaining "common knowledge" about the company. We make use of GoogleSets[7] to get words closely related to the company name. This helps us identify companies similar to the company under consideration, we get to know the products, competitor names etc. This kind of information is very useful, especially for twitter streams, as many tweets compare companies and their products with the competitors. We could for example associate Mozilla, Firefox, Internet Explorer, Safari keywords to Opera Browser entity from the keywords inferred from this source.

UserFeedback Positive Keywords The user himself enters the keywords which he feels are relevant to the company. The keywords we get from the user are of high quality, though they would be few in number. In case of companies where sample ground truth is available, we can infer the keywords from the tweets (in the training set) belonging to the company.

UserFeedback Negative Keywords The knowledge of the common entities with which the current company entity could be confused, would be a rich source of information, using which one could classify tweets efficiently. The common knowledge that "apple" keyword related to "Apple Inc" company could be interpreted possibly as the fruit, or the New York city etc. This particular set of keywords helps us to collect all the keywords associated with other entities with similar keyword. An automated way of collecting this information would be very helpful, but it is difficult. For now we make use of few sources as an initial step to collect this information. The user himself provides us with this information. Second, the wiki disambiguation pages[8] contains this information, at least for some entities. Finally this information could be gathered in a dynamic way i.e., using the keywords in all the tweets, that do not belong to the company. In fact, our more sophisticated classifier to be discussed in section 4.4 exploits this information. The unrelated keywords could also be obtained if we have training set for a particular company with tweets that do not belong to the company entity. Only keywords from this source contribute towards the negative evidence during the classification of tweet.

Table 4.2 shows the basic profile of "Apple Inc"[9] company entity.

We associated a weight proportional to the quality of the source from which these words are extracted. More generally, if a training set is available one can use more sophisticated techniques. From the training set of the company, for each word, let N_r be the number of tweets containing this word and belong to the company. Similarly N_{nr} be the number of tweets in the training set containing this keyword but do not belong to the company. The weight of the keyword can be chosen proportional to $\frac{N_r}{N_r+N_{nr}}$. In this process, there could be many keywords in the profile, where there are no tweets in the training set containing these words. For all such words one can associate a weight proportional to the quality of the

[6] http://wordnet.princeton.edu/

[7] http://labs.google.com/sets

[8] http://en.wikipedia.org/wiki/Apple_(disambiguation) page contains apple entities

[9] http://www.apple.com

Positive Evidence Keywords
HomePage Source: iphone, ipod, mac, safari, ios, iphoto, iwork, leopard, forum, items, employees,itunes, credit, portable, secure, unix, auditing, forums, marketers, browse, genius, music, recommend, preview, type, tell, notif, phone, purchase, manuals, updates, fifa, 8GB, 16GB, 32GB …
Metadata Source: {empty}
Category Source: opera, code, brainchild, movie, trade, paper, freight, keyboard, merchandise, disk, language, microprocessor, move, web, monitor, show, instrument, board, lade, digit, shipment, food, cpu, moving-picture, fluid, consign, contraband, electronic, volume, peripherals, crt, resolve, yield, server, micro, magazine, telecommunications, manage, commodity, flick, vehicle, set, creation, procedure, consequence, second, design, result, mobile, home, processor,spin-off, wander, analog, transmission, cargo, expert, record, database, tube,payload, state, estimate, intersect, internet, print, machine, deliver, job, output, release
GoogleSets Source: itunes, intel, belkin, 512mb, sony, hp, canon, powerpc, mac, apple, iphone, ati, microsoft, ibm
UserFeedback Source (Positive): iphone, ipod, itouch, itv, iad, itunes, keynote, safari, leopard, tiger, iwork, android, droid, phone, app, appstore, mac, macintosh
Negative Evidence Keywords
UserFeedback Source (Negative): fruit, tree, eat, bite, juice, pineapple, strawberry, drink

Table 4.2: Apple Inc. Basic Profile

source from which these words are extracted, as in our simple case. This default weight for the keywords not present in the training set tweets, is similar to default weights usually used for an improved Naive Bayes Classifiers [KRYL02].

4.4 Improved techniques

4.4.1 Relatedness-based Classification

Based on the training set of size 50 tweets per company, we estimate the $relatedness$ factor of a company. We define this term as the percentage of tweets that really belong to the company.

$$relatedness = \frac{\text{\# of tweets in Training Set} \in \text{Company}}{\text{\# of tweets in the Training Set}} \tag{4.9}$$

Figure 4.1 shows the estimated $relatedness$ factor of the different companies in the test set. Companies with higher $relatedness$ factor (for example: Sony, Starbucks, MTV etc.), implies majority of the tweets containing the company keyword belong to the company. Similarly for companies with very low $relatedness$ factor (for example: Seat, Orange, Camel etc.), implies the majority of the tweets mentioning the company keyword do not refer to the company. Note that the $relatedness$ factor characterizes a company based on the dataset and it is independent of the entity profiles.

When classifying a tweet, we actually compare the words present in the tweet against the words present in the profile of a company. Since the number of words we have in the profile are often limited

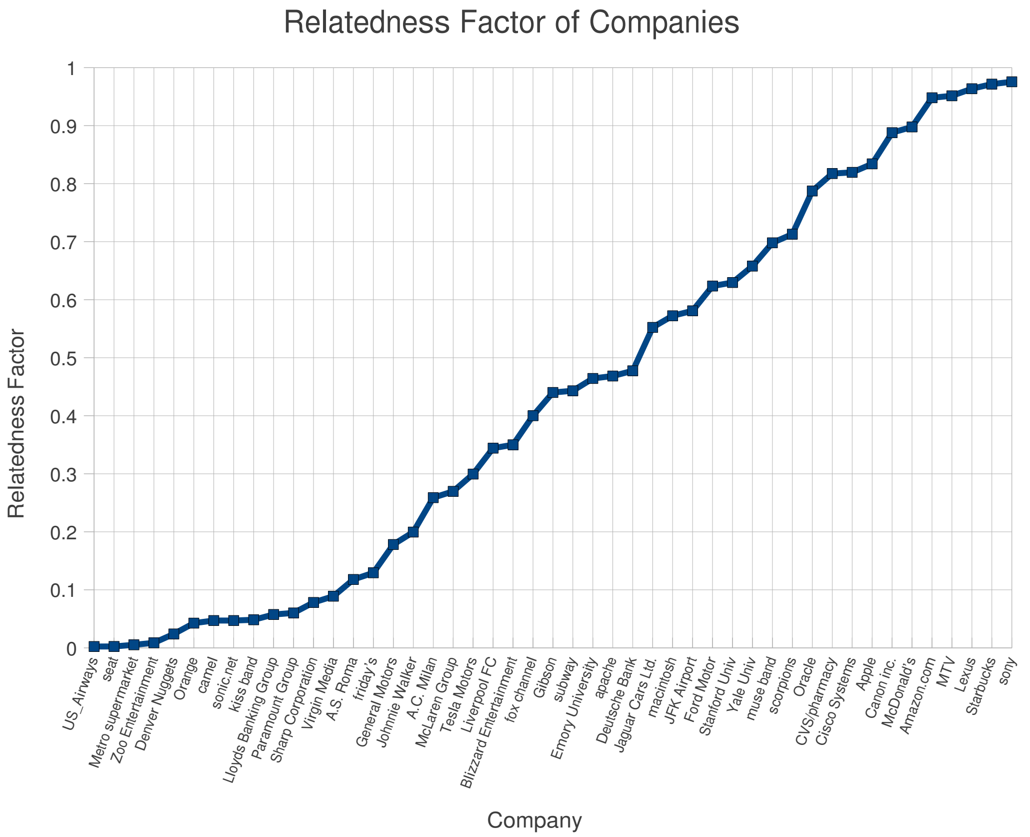


Figure 4.1: Relatedness Factor of Companies

and the possible set of words present in tweet is potentially infinite, in many cases, for many tweets, we do not find any overlap with the company profile. In such cases, it would be better to classify such tweets according to the $relatedness$ factor of the company. The knowledge of the $relatedness$ factor helped us to improve the accuracy of our classification. This technique particularly improves the performance in the cases, where the constructed company profiles are small or have low quality.

Once we know (i.e. estimate) the $relatedness$ factor of a company, there are two ways of classifying an unseen tweet. The first strategy is, if this factor is greater than 0.5, for all tweets we classify them as belonging to the company. This way of classifying helps us achieve an expected accuracy equal to the $relatedness$ factor. When the $relatedness$ factor of a company is less than 0.5, all the tweets are classified as not belonging to the company. In this case, we achieve an expected accuracy of 1 - $(relatedness)$.

The second way is, for each tweet we classify the tweet belonging to the company with a probability equal to the $relatedness$ factor. In this way of classification, we would have tweets in both the classes: belonging to the company and not belonging to the company. The expected accuracy of this process can be shown to be a little lower than first case, but we gain some knowledge in this probabilistic classification which could be used for classifying future unseen tweets. We explain in more detail how we can infer some useful information using this method in the following section (Section 4.4.2).

Let us denote by N the number of tweets to be classified. With $p = relatedness$ factor, we have $p \times N$ tweets belonging to the company and $(1 - p) \times N$ tweets not belonging to the company. When we decide with probability p that a tweet belongs to the company, we would be right with $p^2 \times N$ tweets as belonging to the company and $(1-p)^2 \times N$ tweets as not belonging to the company. So, in total the expected accuracy is given as:

$$\text{Expected Accuracy} = p^2 + (1-p)^2, \text{where } p \ = \ relatedness\text{-factor.} \qquad (4.10)$$

We assume that the $relatedness$ factor of a given company does not change in time. We can make this assumption as these changes are relatively slow. One can observe dynamic changes of individual word frequencies which we handle using a different technique, that we explain in the next section.

4.4.2 Active Stream Learning Based Classification

In Section 4.3 we described how we constructed a basic profile of the company using few reliable sources (such as company homepage, category keywords, Google sets keywords, user feedback etc.) which give us list of keywords which help us decide if a tweet belongs the company. The basic profile is a good starting point for building an efficient classifier, however there are severe limitations of just using the basic profile, which we need to address in order to design better classifiers. In this section, we identify these limitations and propose novel techniques to overcome them.

The efficiency of the basic profile is limited by number of tweets in the test set that contain at-least few overlapping words from the basic profile. From the analysis of the test set tweets we observe that there is a significant percentage of tweets, which do not have any overlapping words with the corresponding basic profile keywords. The Figure 4.3 in Experiments section confirms this observation.

Some of the limitations of using only the basic profile include:

1. The number of keywords in the basic profile are limited, while the number of words one could find in a twitter stream of the company are potentially infinite.

2. The sources from which we gather the basic profile keywords are good for collecting positive evidence keywords but not so good for negative evidence keywords. It is possible, at least through human input and with the help of many Web sources, to associate all possible keywords related to a company. On the other hand it is relatively difficult to get a list of entities with which a company keyword could be confused. There is no single authoritative source on the web which lists all possible interpretations of a company name.

3. The basic profile does not consider the characteristics of the words distribution in a tweet stream. The power law shown by word frequencies of tweet words, suggests which words should be present in the company profile so as to make an intelligent decision.

4. The $relatedness$ factor of a company is useful information, which is completely ignored by a classifier that solely relies on the basic profiles.

5. The limited user feedback is completely ignored by the basic profile. Usually it is difficult to involve humans in classifying the tweets, as there are numerous tweets in amount. Even for some number of tweets for which the user is willing to provide feedback, is not exploited by the basic profile.

Algorithm 4.1: Active Stream Learning

```
Input : Basic Profile: $P_0.Set^+, P_0.Set^-$
 Twitter Stream: $\Gamma = \{T_1, \ldots, T_n\}$
 R : $Relatedness$ factor of company
Init : Active Tweet Sets: $P_\triangle.Set^+ = \{\}, P_\triangle.Set^- = \{\}$
for all $T_i \in \Gamma$ do
  $score$ = SCORE($T_i, P_0.Set^+$) + SCORE($T_i, P_0.Set^-$)
  if $score > 0$ then
    $P_\triangle.Set^+$.add($T_i$,score)
  else if $score < 0$ then
    $P_\triangle.Set^-$.add($T_i$,score)
  else
    if $Math.radom(0, 1) < Relatedness$ factor then
      $P_\triangle.Set^+$.add($T_i$,$Relatedness$)
    else
      $P_\triangle.Set^-$.add($T_i$,$Relatedness$)
    end if
  end if
end for
{ $P_\triangle.Set^+$,$P_\triangle.Set^-$ } = WordFreqAnalysis($P_\triangle.Set^+, P_\triangle.Set^-$)
Add Top-K keywords or all words above Threshold from $P_\triangle.Set^+$ to $P_0.Set^+$
Add Top-K keywords or all words above Threshold from $P_\triangle.Set^-$ to $P_0.Set^-$
return $P_0.Set^+, P_0.Set^-$
```

Few observations made on the twitter streams, along with identifying $relatedness$ factor of the company helps us in overcoming many limitations of the basic profile based classifier. Here we discuss our observations and how we make use of them in developing more accurate classifier.

For each company we inspected the messages from the twitter stream which contain the given company name as a search keyword. For each company, by inspecting the twitter stream [10] (of about 2000 tweets), we studied the word frequency distributions. In general, we could observe power law of distributions for word frequencies. If we have a knowledge about all or top-k of these words, and if these words contribute as positive or negative evidence, then this should help us in classifying many more tweets from test set more accurately. Indeed, we applied such techniques.

The premise we use for improving over basic profile classifier is, to add more words to the positive and negative evidence profile. While adding these words we have to make sure they are of high quality and if they have more possibility of appearing in the future tweets. Some of the tweets which we are able to identify accurately using the basic profile, provide us more keywords, which can be used to resolve new unseen tweets. For example, assume our basic profile about Apple Inc. company contained only keywords {iPhone, iPod, mac}. Now when inspecting tweets from stream containing the "apple" keyword, we observe that there are many tweets mentioning "iPhone" and "iPad" together. Since we are able to classify all such tweets as belonging to the Apple Inc. company by the virtue of "iPhone" keyword, we can confidently associate "iPad" word also as a useful word which helps us associate future tweets containing only "iPad" keyword as belonging to Apple Inc.

[10] `http://search.twitter.com/search.json?q=COMPANY-NAME`

As discussed in Section 4.3, in our representation the basic profile contains two sets of weighted keywords. The set with positive weights contribute as positive evidence while the negative weights set contribute as negative evidence. The weights of the words signify how confident the word helps in classifying the tweet as belonging to or not belonging to the company.

We proceed as follows (Algorithm 4.1). We start inspecting the twitter stream using this basic profile. Of the many tweets we inspect some percentage of tweets, which have overlap with the basic profile, can be accurately classified. All words co-occurring with profile keywords in these tweets can be added to the profile. The weights we associate with these newly identified keywords should depend on the words which made them as possible candidates and also on number of times they co-occurred.

Also when inspecting twitter stream, we would come across many tweets which do not have any overlap with the basic profile keywords. For all such tweets, we classify based on the $relatedness$ factor of the company. We end up with two sets of tweets: one set of tweets which we classify as belonging to the company and the other set as not belonging to the company. For both the sets, based on the word frequency distribution, we add all the keywords above certain threshold to the profile. The weight we associate with these words should depend on number of times the word appears and the $relatedness$ factor.

When there is feedback on some of the tweets by the user, this model is able to use the feedback very efficiently. All the tweets on which the user has responded, the active stream learning algorithm can ignore the basic profile-based and $relatedness$ factor-based decisions and give more weight-age to the user responded tweet keywords.

4.5 Experimental evaluation

Experimental setup

We performed our experiments on a 2GB RAM, Genuine Intel(R) T2500 @ 2.00 GHz CPU. Linux Kernel 2.6.24, 32-bit machine. We implemented our methods using matlab, java and python.

Dataset

We used the WePS-3 Dataset available here[11] as our test set. This dataset contained about 47 companies, with each company having about 450 tweets. All the tweets corresponding to a company are annotated as belonging to or not belonging to the company. For each company we randomly selected 50 tweets out of about 450 tweets as our training set. We used the training set only for estimating the $relatedness$ factor for each company. For constructing the active profiles, we gathered twitter streams for each company, using the query term shown in companies dataset table in [AAG$^+$10], from Twitter search API[12]. The number of tweets we investigated for active profiles varied from 600 to 9900 tweets.

Metrics

The task is of classifying the tweets into two classes: one class which represents the tweets related to the company (positive class) and second class represents tweets that are not related to the company (negative class). For evaluation of the task, the tweets can be grouped into four categories: true positives (TP),

[11] http://nlp.uned.es/weps/weps-3/data

[12] http://search.twitter.com

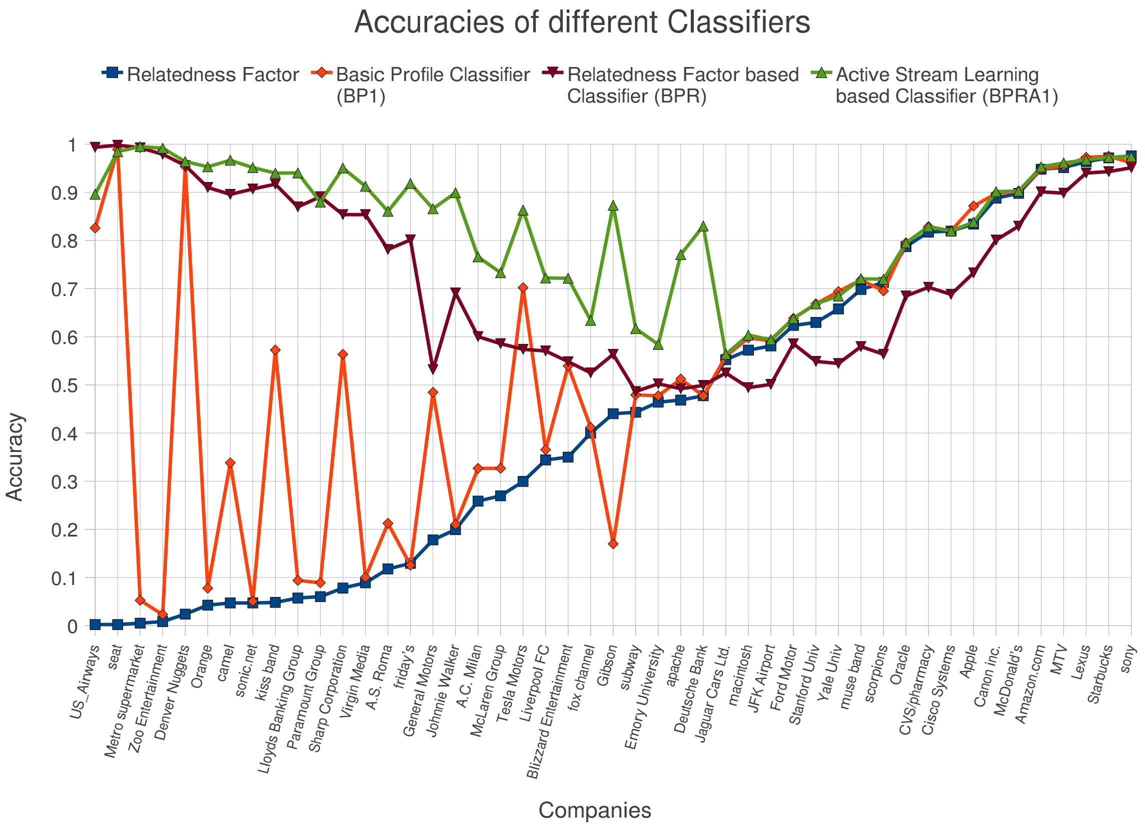


Figure 4.2: Accuracies of different Classifiers

true negatives (TN), false positives (FP) and false negatives (FN). The true positives are the tweets that belong to positive class and in fact belong to the company and the other tweets which are wrongly put in this class are false positives. Similarly for the negative class we have true negatives which are correctly put into this class and the wrong ones of this class are false negatives.

We use the *accuracy* metric to study the performance of our different classifiers.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.11}$$

Different Classifiers

Our experiments make use of following different classifiers:

1. Basic Profile-based Classifier (BP1): For each company we formed the basic profile, which included keywords from all the sources: homepage, category, metadata, google sets and user feedback.

2. Basic Profile-based Classifier (BP2): In general we observed that keywords extracted from homepage source are of low quality compared to all other sources. So, we formed a second basic profile whose keywords are from high quality sources like category, metadata, google sets and user feedback.

3. $Relatedness$ factor based Classifier (BPR): Based on the training set we estimated the $relatedness$ factor of each company. Using this factor the classifier classified all the tweets.

4. Active Profile Classifier (BPRA1): We used high quality basic profile (BP2), which considered only high quality sources, for forming the active profile. This classifier based on the active profile classified all the tweets in the test set.

5. Active Profile Classifier (BPRA2): In order to study the impact of the quality of basic profile on the construction of active profile, we used basic profile (BP1) for forming the active profile. This classifier based on the active profile (BPRA2) is used to classify all the tweets in the test set.

6. Active Profile Classifier (BPRA3): We earlier discussed that the quality of the active profile depends on how good the starting basic profile we use for its construction. For the active profile classifier BPRA3 we assume that the initial basic profile is empty, and go about constructing the active profile based only on the $relatedness$ factor decisions.

Please note that the classifiers (BPRA1), (BPRA2) and (BPRA3) internally make use of the estimated $relatedness$ parameter, as it is explained in Algorithm 4.1.

In the first set of experiments, we study how the different classifiers performed on the test set. The accuracy metric of the different classifiers : BP1, BPR and BPRA1 are shown in the Figure 4.2. We see that on average the $relatedness$ factor based classifier (BPR) and active profile based classifier (BPRA1) outperform the basic profile-based classifier (BP1). Also the BPRA1 classifier outperformed BPR classifier. On close observation of the Figure 4.2, we see that for the companies on the far-right that is with high $relatedness$ factor, the profile-based classifiers BP1 and BPRA1 are better than the classifier BPR. The reason is, the basic profile is already good enough to capture all the useful words associated with the company. The active profile does not improve much on the basic profile. Thus they both outperform the classifier (BPR). This is in tune with the argument in Section 4.4 that it is relatively easy to gather positive evidence keywords compared to the negative evidence keywords.

In the left side of the graph where the relatedness factor of the companies are low, we observe that BPR and BPRA1 clearly outperform BP1. It strongly suggests that the basic profile was not good enough to contain all the negative evidence keywords associated with the company. BPR is outperforming because it is exploiting the $relatedness$ factor estimate. While BPRA1 was able to efficiently identify all the supporting keywords which were not initially available in the basic profile.

The significant performance improvement of active profile-based classifier over the basic profile based classifier can be attributed to the fact that the active profile is able to identify many more keywords just by inspecting the twitter streams. In Figure 4.3 we show number of words in the profiles that overlap with the top 50 keywords of the test set. It confirms our observation that only small percentage of tweets in the test set overlap with the keywords in the basic profile. We also see that by use of active profile, there is significant percentage of overlap between the keywords in the test set and the active profile.

The quality of the active profile we construct depends on the quality of the basic profile that is used. In order to study how the different basic profiles affect the active profile based classifiers performance, we constructed many active profiles BPRA1,BPRA2 and BPRA3, each starting with a different quality basic profile. From the description of the different basic profiles, we see that the quality of BP2 classifier is better than BP1 classifier, which further are better than the empty basic profile. The average performance

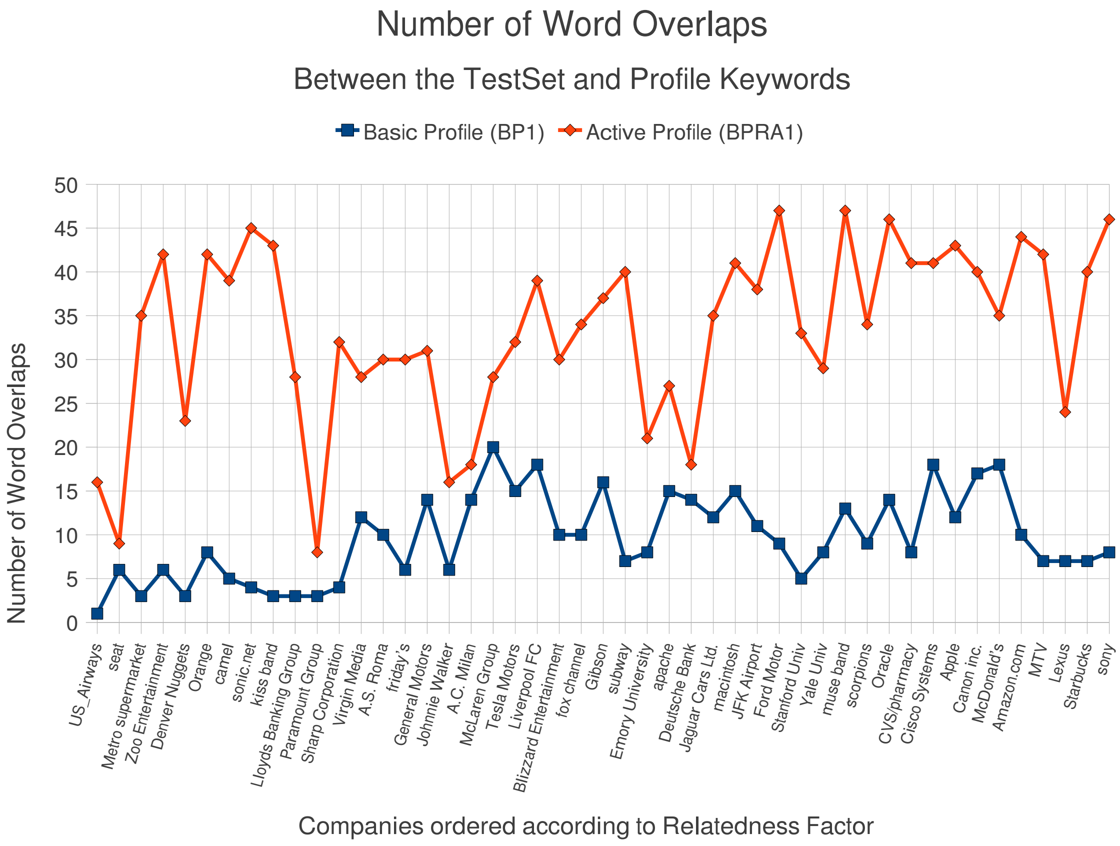


Figure 4.3: Number of word overlaps between a company profile and corresponding tweets test dataset.

of each of the different classifiers is shown in the Table 4.5. From the table we observe that BPRA1 is better than BPRA2 which in turn is better than BPRA3 classifier. Thus we observe that as the basic profile quality deteriorates so does the performance of the corresponding active profile.

4.5.1 Performance of the Classifiers in WePS-3 ORM Task

A number of teams participated in WePS-3 Online Reputation Management (ORM) Task, concerning classification of tweets containing the company name if they are related to the company entity. The participating teams submitted multiple classifiers for the task. We participated in this challenge with a classifier that is based on BP1 (Basic Profile using all sources) and $Relatedness$ factor, which we refer to as *LSIR.EPFL#1* classifier henceforth. Other competing classifiers were from KALMAR [Kal10], SINAI [CVSPO10], UVA [TB10], and ITC-UT [YMO$^+$10]. We discuss their approaches in Section 4.7.

Table 4.4 shows the performance of the different classifiers based on a number of performance metrics. The table also shows two baseline classifiers: $BASELINE_R$ which classifies each tweet as related to the company , while $BASELINE_{NR}$ classifies each tweet as not-related to the company. We see that our classifier, *LSIR.EPFL#1*, is performing the best on a number of performance metrics and mainly in the overall accuracy and related-ratio-deviation. We also see that the baseline classifiers ($BASELINE_R$ and $BASELINE_{NR}$) are also performing best on some of the performance metrics. It is not clear how to unanimously rank all the participating classifiers. F-measure is just one way of combining precision and recall measures by giving equal importance to each of them.

Classifier	Average Accuracy
Basic Profile using all sources (BP1)	0.43
Basic Profile using only high quality sources (BP2)	0.46
$Relatedness$ factor based classifier (BPR)	0.73
Active Profile constructed using high quality Basic Profile-BP2 (BPRA1)	0.84
Active Profile constructed using normal quality Basic Profile-BP1 (BPRA2)	0.79
Active Profile constructed using the empty Basic Profile (BPRA3)	0.76

Table 4.3: Average Accuracy of Different Classifiers

The judges of the WePS-3 task relied on Unanimous Improvement Ratio (UIR) measure [AGAV11] to rank all the participating classifiers based on pairwise relative performance. In the UIR measure they vary the relative importance of precision and recall, and study in the number of cases in which one system performs better than the other. Table 4.5 shows which classifiers improve on which other classifiers. It can be observed that *LSIR.EPFL#1* has shown improvement over most of the other participant classifiers. Also observe that the baseline classifiers ($BASELINE_R$ and $BASELINE_{NR}$) , despite having good performances over few metrics, are not improving on any other classifiers.

4.6 Performance Analysis and Further Improvements

We have introduced and evaluated various Twitter classification methods. In Section 4.3 we started with a simple classifier only relying on a basic profile, while in Section 4.4 we improved this method through the use of the $relatedness$ factor and updates from the active Twitter stream. In Section 4.5 we evaluated these methods. Our evaluation shows that the performance of these classifiers is still leaves some room for improvement, for some companies. In this section we look into the reasons for the under-performance and also propose principled techniques for improvements.

As a summary, our classifiers work as follows. A company profile in our setting is a set of weighted keywords. When a company profile is used for classifying an unseen tweet, the Naive-Bayes classification looks for overlapping keywords in the tweet message and in the company profile. The net sum of the weights of the overlapping words, determines if the tweet belongs to the company or not. For all the tweets which do not have any overlapping words with the profile, we classify those tweets based on the $relatedness$ factor of the company.

We first introduce some useful concepts for studying the performance of a classifier. The performance of a classifier, given a company profile on the test set collection of tweets, depends on how well the keywords of the test set collection overlap with the company profile keywords and how accurate are the weights in the company profile. Thus, to improve the performance of classifiers, we need "better" profiles, that is profiles that contain a high number of relevant keywords which also appear in the test set collection, with as accurate weights as possible.

For our performance analysis, we define following concepts:

Run	Non Processed Tweets	Precision (related)	Recall (related)	F-measure (related)	Precision (non related)	Recall (non related)	F-measure (non related)	Accuracy	Related Ratio Deviation
LSIR.EPFL#1	0	0.71	0.74	**0.63**	**0.84**	0.52	0.56	**0.83**	**0.15**
ITC-UT#1	0	0.75	0.54	0.49	0.74	0.6	0.57	0.75	0.18
ITC-UT#2	0	0.74	0.62	0.51	0.74	0.49	0.47	0.73	0.23
ITC-UT#3	0	0.7	0.47	0.41	0.71	0.65	0.56	0.67	0.26
ITC-UT#4	0	0.69	0.55	0.43	0.7	0.55	0.46	0.64	0.32
SINAI#1	449	0.84	0.37	0.29	0.68	0.71	0.53	0.63	0.36
SINAI#4	449	0.9	0.26	0.17	0.73	0.72	0.53	0.61	0.38
$BASELINE_{NR}$	0	1	0	0	0.57	1	0.66	0.57	0.43
SINAI#2	449	**1**	0	0	0.58	**0.98**	**0.65**	0.56	0.43
UVA#1	409	0.47	0.41	0.36	0.6	0.64	0.55	0.56	0.27
SINAI#5	449	0.72	0.51	0.28	0.75	0.47	0.33	0.51	0.48
KALMAR#4	874	0.48	0.75	0.47	0.65	0.25	0.28	0.46	0.43
SINAI#3	449	0.6	0.7	0.36	0.86	0.28	0.19	0.46	0.54
KALMAR#2	874	0.47	0.7	0.43	0.61	0.27	0.28	0.44	0.43
KALMAR#5	874	0.48	**0.77**	0.47	0.65	0.21	0.23	0.44	0.45
$BASELINE_{R}$	0	0.43	1	0.53	1	0	0	0.43	0.56
KALMAR#1	2207	0.51	0.7	0.42	0.59	0.19	0.21	0.4	0.39
KALMAR#3	2202	0.49	0.66	0.39	0.66	0.25	0.27	0.4	0.47

Table 4.4: Final Ranking of the Classifiers that participated in WePS-3 Online Reputation Task Challenge [AAG+10]

Perfect Profile : P_c : We define the Perfect Profile, P_c, of a company as the profile that can be formed using the words inferred from the entire test set. The weights associated with these words reflect the distribution of words in the entire test set collection.

Eventually, when one uses this profile for classifying the tweets in the test set, with the given classification method we will have the best possible performance. The performance of the classifier that uses the Perfect Profile is an upper bound for the accuracy level of the classifier with any other profile.

Current Profile : P_i : It is the profile that is formed using the different techniques proposed in the earlier sections (Sections 4.3 and 4.4) that is eventually used by the classifier.

Next we look into the performance differences of the Current Profile P_i w.r.t. the Perfect Profile P_c

4.6.1 Comparison of the Current Profile and the Perfect Profile of a Company

We summarize the performance of Current Profile in relation to the performance of Perfect Profile in the Figure 4.4. We observe that Current Profile is doing as good as Perfect Profile for the companies with either very low or very high $relatedness$ factor. For these companies, the Current Profile is able to capture the required keywords accurately using the mentioned techniques. However, the Current Profile still lags behind Perfect Profile for the companies with mid-range $relatedness$ factor. If we want to further improve the classification performance, we need to look into the reasons for the under performance of Current Profile for companies with mid-range $relatedness$ factor.

In Figure 4.5, we show the comparison of Perfect Profile against Current Profile of a mid-range $relatedness$ factor company (Company name: "Emory University"). The words on the x-axis are arranged in a decreasing order of their occurrence frequency in the test-set collection. The top graph shows

Run	Accuracy	Improved Systems
LSIR.EPFL#1	0.83	KALMAR#1, KALMAR#5, ITC-UT#2, KALMAR#2, KALMAR#3, ITC-UT#4, KALMAR#4, UVA#1, $BASELINE_{NR}$
ITC-UT#1	0.75	SINAI#4, UVA#1
ITC-UT#2	0.73	SINAI#4, UVA#1
ITC-UT#3	0.67	KALMAR#2, KALMAR#3, #UVA#1
ITC-UT#4	0.64	SINAI#4, UVA#1
SINAI#1	0.63	SINAI#4, SINAI#2, UVA#1 $BASELINE_{NR}$
SINAI#4	0.61	
$BASELINE_{NR}$	0.57	
SINAI#2	0.56	
UVA#1	0.56	KALMAR#1, KALMAR#2, KALMAR#3
SINAI#5	0.51	
KALMAR#4	0.46	KALMAR#1, KALMAR#5
SINAI#3	0.46	
KALMAR#2	0.44	
KALMAR#5	0.44	
$BASELINE_{R}$	0.43	
KALMAR#1	0.4	$BASELINE_{NR}$
KALMAR#3	0.4	KALMAR#1

Table 4.5: UIR results (UIR threshold=0.1). Relative performance of the Classifiers that participated in WePS-3 Online Reputation Task Challenge [AAG+10]

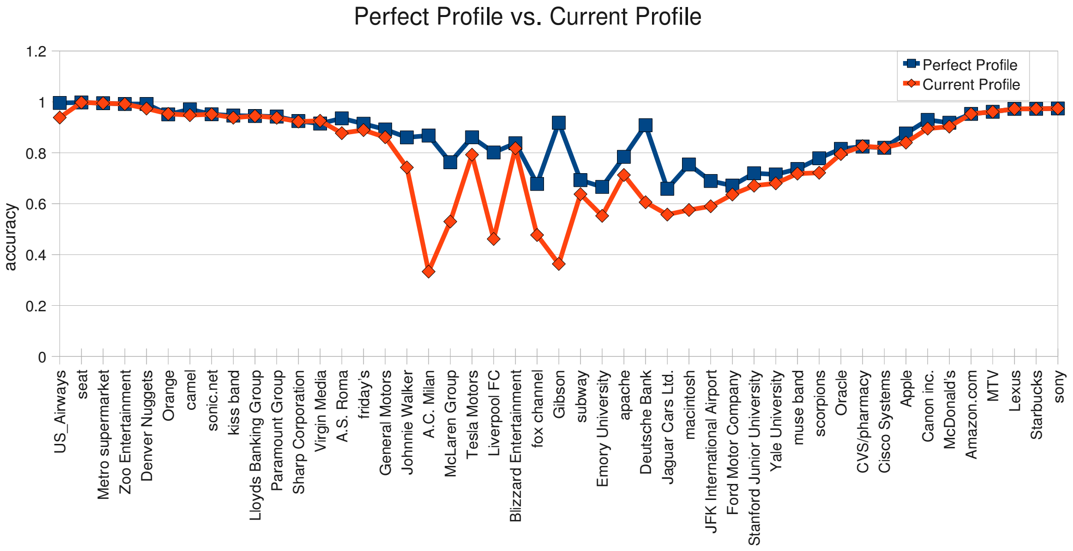


Figure 4.4: Comparison of accuracies of Current Profile vs. Perfect Profile.

the Perfect Profile, with grey-bars referring to the positive weights and black-bars referring the negative weights. The height of the bars indicate associated weight. The lower graph represents the Current Profile of the company. Once again the grey-bars and black-bars indicate positive and negative weights respectively. The reverse-slashed-hatched-bars indicate positive weights but their corresponding weights in the Perfect Profile is negative. Similarly horizontally-hatched-bars indicate negative weights while their corresponding weights in the Perfect Profile is positive. These hatched bars in a way contribute towards the reduced performance of the classifier.

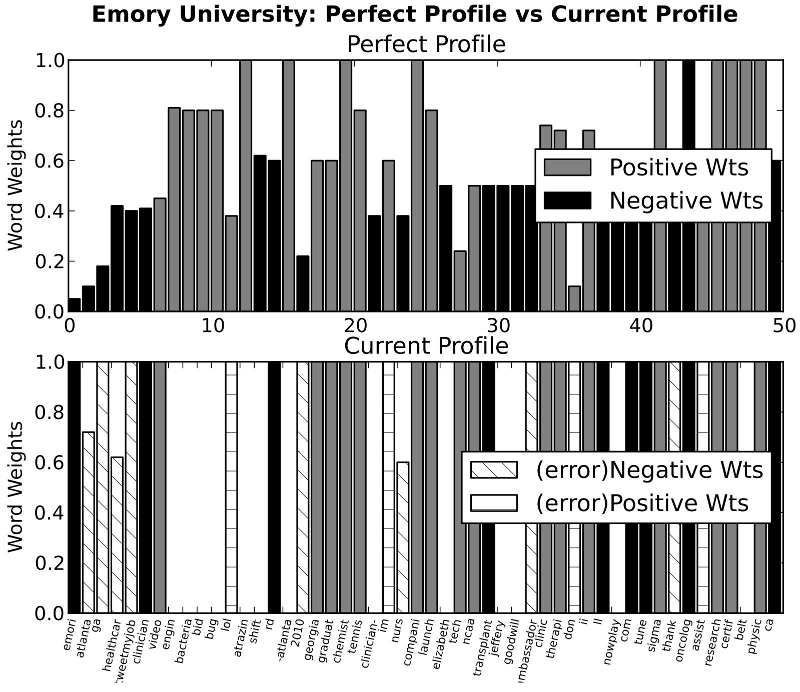


Figure 4.5: Comparison of a company's profiles (Current Profile vs. Perfect Profile)

Figure 4.5 helps us understanding the possible reasons for the under-performance of Current Profile in comparison to the Perfect Profile. First, we observe that there are certain words in Perfect Profile, whose corresponding weights in the Current Profile is zero. The Current Profile does not contain any information about these words that are occurring in the Perfect Profile. The reason could be that, when the profile is constructed, those words are not encountered. So, the Current Profile will not be able to classify the tweets containing those words accurately. Second, we observe some words acting as "positive evidence" (i.e. information indicating that the keyword in the message is related to the company) in Perfect Profile are acting as "negative evidence", indicated by the horizontally-hatched-bars, and similarly some words acting as "negative evidence" in Perfect Profile are acting as "positive evidence", indicated by the reverse-slashed-hatched-bars. All such words also contribute to some error in the classification. Thirdly, there could be an error because of differences in the weights of words in the Perfect Profile and the Current Profile.

4.6.2 Error Groups

On comparing the Current Profile with Perfect Profile, we have seen the different ways in which the errors could occur. Based on the observations we define three different error groups as follows.

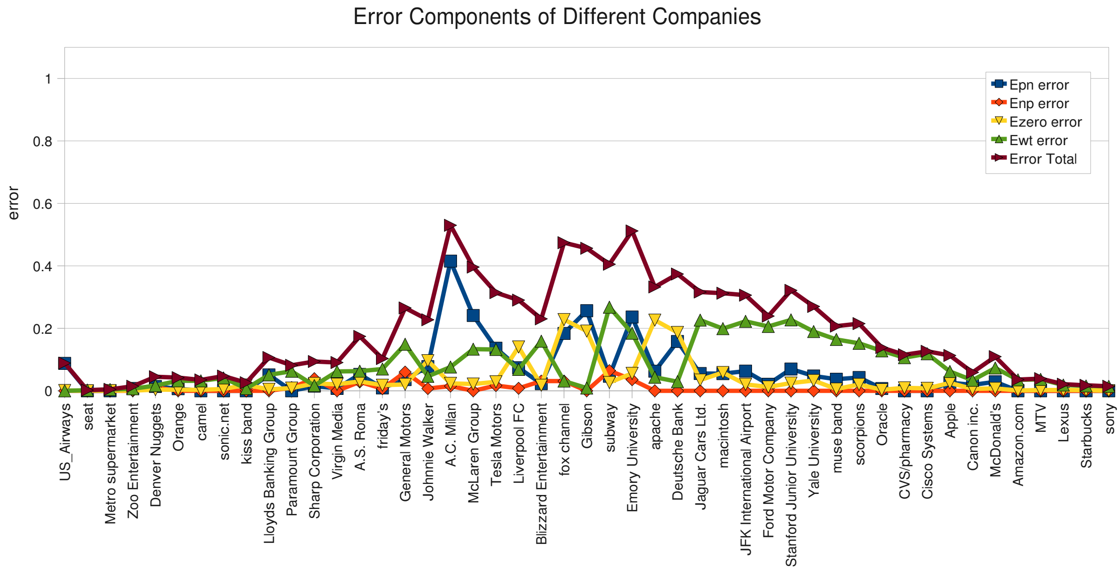


Figure 4.6: Different error components contributing towards total classification error

Missing Words Error: (E_{zero}): The Current Profile, under consideration, may not contain some words appearing in the Perfect Profile, i.e. the frequent words that are appearing in the test-set collection. The classifier with the Current Profile in this case would classify all such tweets using the $relatedness$ factor of the company. In this case, the classification error occurs because of these $relatedness$ factor-based decisions. We denote the fraction of incorrect decisions of this type as E_{zero}, that can be computed as follows:

$$E_{zero} = \sum_{i} (1 - relatedness) \left(\frac{\text{\# of Tweets containing } wrd_i}{\text{\# of Tweets in Test Set}} \right) \tag{4.12}$$

where wrd_i are the missing words i.e., the words which appear in Perfect Profile but not in Current Profile.

Wrongly Placed Words Error: $E_{PN}(E_{NP})$ is the error caused because of words, which are supposed to be acting as positive (negative) evidence are instead of acting as negative (positive) evidence. The Current Profile classifies all such tweets containing this misplaced words with a confidence proportional to the weights of the misplaced words. So the error introduced will be proportional to the weights of the misplaced words.

$$E_{NP} = E_{PN} = \sum_{i} \left(\frac{1 + \|wt_i\|}{2} \right) \left(\frac{\text{\# of Tweets containing } wrd_i}{\text{\# of Tweets in Test Set}} \right) \tag{4.13}$$

where wrd_i are the misplace words i.e, words which are acting as positive (negative) evidence in active-profile are acting as negative (positive) evidence in Current Profile, and wt_i is the weight of the wrd_i in Current Profile.

Words Weights Error: E_{wt} is the error caused because of the differences in the weights of words in the Current Profile and the Perfect Profile. The tweets containing these words (wrd_i) are classified with a confidence of wt_i, weight associated with the word in Current Profile, instead of a confidence of wt_i^p, the weight associated with the word in Perfect Profile.

$$E_{wt} = \sum_i \left(\frac{\|wt_i - wt_i^p\|}{2} \right) \left(\frac{\text{\# of Tweets containing } wrd_i}{\text{\# of Tweets in Test Set}} \right) \tag{4.14}$$

The above described different error groups, for all the companies, are shown in Figure 4.6. We see that the majority of the errors is in the middle of the graph, corresponding to the companies with mid-range $relatedness$ factor. We can further see the different components: E_{zero}, E_{PN}, E_{NP} and E_{wt} contribution towards the total error.

4.6.3 Reducing the Error Components

In this section we discuss methods and tradeoffs for reducing errors (of different types defined in Section 4.6.2).

4.6.3.1 Reducing the Missing Words Error (E_{zero})

We have seen that we construct the profiles using the static information sources (for example, homepages, etc.), that we then extend with keywords from the active twitter streams. This learning mechanism helps us increase the overlap of words between the Current Profile and the Perfect Profile. It is natural that the longer we inspect the active twitter stream, the higher is the probability of learning new words. Thus the size of the active stream that we inspect, has direct impact on the number of new words that we include in the profile.

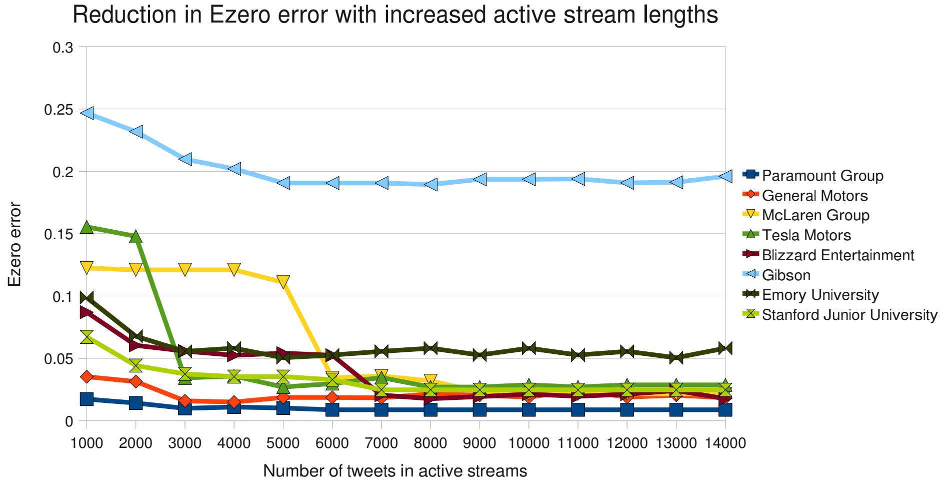


Figure 4.7: Reduction of Missing Words Error (E_{zero}) component of selected companies

When the Missing Words Error E_{zero} component, is significant we should try increasing the length of active stream of inspection. We conducted an experiment in which we formed the Current Profile using active streams of increasing length (from the size of 1000 to 14000 tweets per company). Figure 4.7 shows the impact on the Missing Words Error (E_{zero}), for some mid-range $relatedness$ factor companies, with the increasing active twitter stream length, we see that the E_{zero} component reduces as the active twitter stream length increases. We observe that even though the error E_{zero} reduces, it

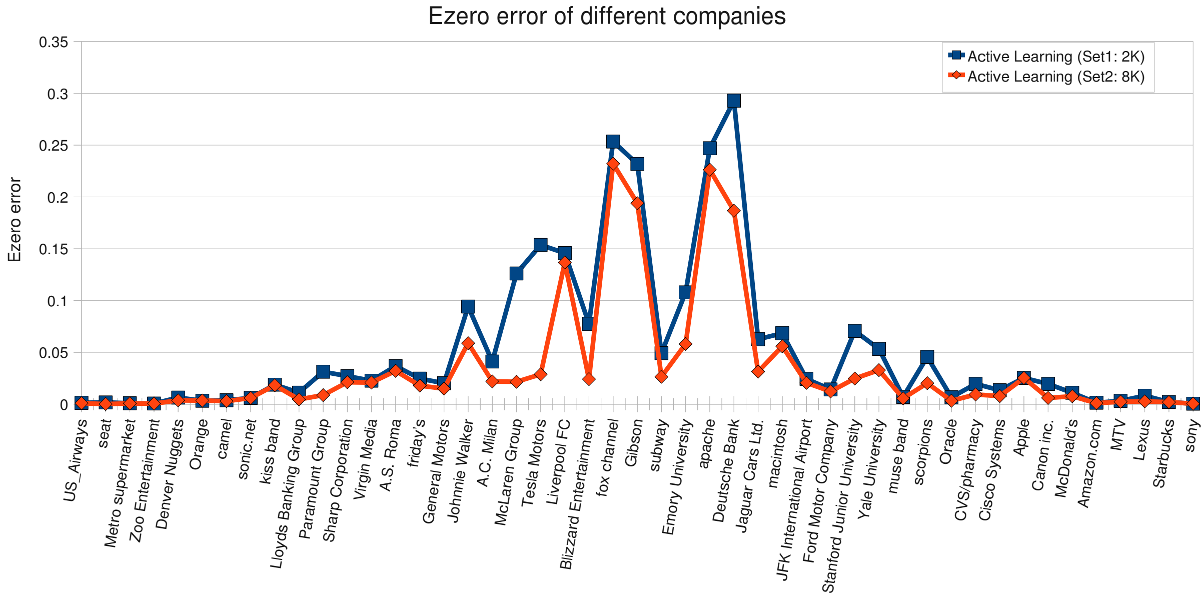


Figure 4.8: Comparison of Missing Words Error (E_{zero}) component of all companies for two sets of active stream tweets

never reduces to absolute zero, implying that inspected twitter streams are not containing the words one is expected to find in the test-set collection.

In the next figure we will show the summarized performance for all the companies. Figure 4.8 shows the reduction in E_{zero} component when the Current Profile uses longer active twitter stream (average length of 8K tweets) instead of a smaller active twitter stream (average length of 2K tweets). We observe the error E_{zero} reduction for the mid-range $relatedness$ factor companies.

4.6.3.2 Reducing the Wrongly Placed Words Error (E_{PN} and E_{NP})

With the previous technique we see that we can increase the overlap of words between the Current Profile and Perfect Profile, but this still does not ensure that we are using the newly found words from the stream correctly. We discuss two possible techniques for reducing the Wrongly Placed Words Error component, with their associated costs.

First, we can make use of stricter controls when deciding if a new word should be acting as positive or negative evidence. We usually identify new words when they are co-occurring with the already existing profile keywords. We can associate a weight for the newly found words, based on the quality of the words which identified them and also how frequently the newly found word is occurring. One can have stricter controls policy for adding keywords to the profiles, for example by only adding those new words whose weight is above certain predefined threshold. In the experiments section we have already shown that starting with high quality profile, we usually make less error with adding the newly collected words. If we chose very strict control, like very high threshold, we may run in the risk of missing many useful new words, which in turn can increase the error E_{zero} component.

Another way of reducing the E_{PN} and E_{NP} error, is to make use of user feedback. We can either make use of user feedback on a selected subset of tweets or on a selected set of frequently occurring keywords. In the remaining of this chapter, we make use of the user feedback. We present a set of keywords to the user, who has to evaluate whether they are related to the given company. We treat

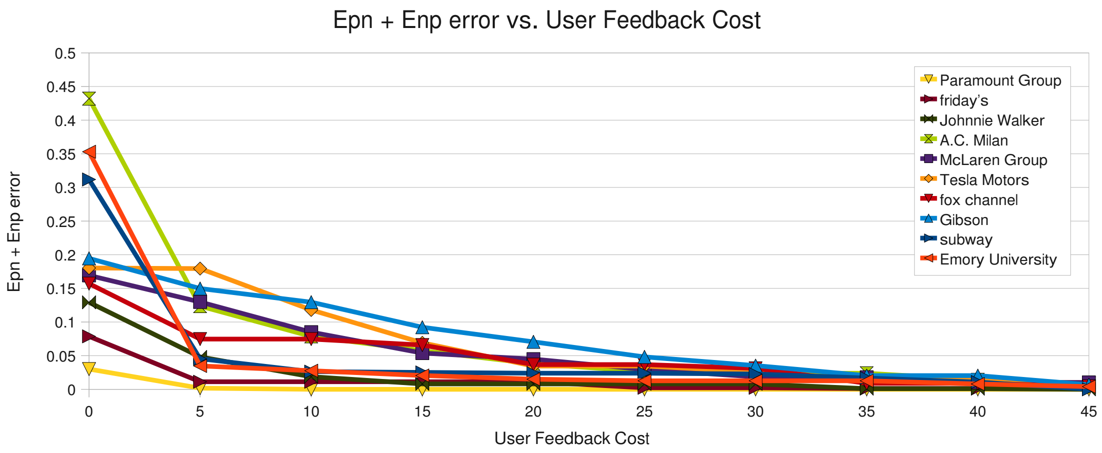


Figure 4.9: Comparison of (E_{PN} and E_{NP}) error component of select set of mid-range $relatedness$ factor companies.

the number of words to which the user gives feedback as the associated human cost. We conduct an experiment in which we study the impact of error E_{PN} and E_{NP} with respect to the user feedback (cost). Figure 4.9 shows the impact on the error E_{PN} + E_{NP}, with the increased cost of user feedback, for some selected set of mid-range $relatedness$ factor companies. We see the error reduces at the expense of user feedback. If we have limited budget of human feedback, we should be careful in choosing only those subset of words which can have maximum impact on the overall performance. This is one of the strength of our approach: based on the error analysis, we can chose only those word which are occurring frequently but whose associated weights are smaller than the chosen threshold. In this way we can “optimally” use the costly human input. (In fact, we did not conduct our experiments with human users directly, rather we considered the ground truth as human input. The ground truth itself was created through human effort, see [AAG$^+$10].)

4.6.3.3 Reducing the Words Weights Error (E_{wt})

While it is clear how to reduce the errors E_{zero} and E_{PN}+E_{NP} by inspecting longer active twitter streams and efficiently using the human feedback, it is really difficult to reduce the error due to differences in the weights of words in the Current Profile and Perfect Profile. The weights are obtained through heuristic techniques (see Section 4.3), as no good training set is available. For reducing this error E_{wt} we could construct a training set that represents well the test set, however obtaining a good training set may be difficult.

4.6.4 Error reduction techniques impact on the overall accuracy performance

After seeing the different ways of reducing the individual error components, now we present the impact on the overall accuracy. The following table shows the accuracy performance of different profiles. As Table 4.6 shows, the error correction techniques explained above further improve the accuracy of our classification techniques. The results using the Current Profile are approaching the ones of Perfect Profile, one could even further improve them, if needed. There are certainly a limitations how close we can get, because of the Words Weights Error component (E_{wt}).

Different Profiles	**Overall Accuracy**
Perfect Profile	0.87
Current Profile	0.79
Current Profile combined with error E_{zero} reduction technique	0.81
Current Profile combined with error E_{pn}+E_{np} reduction technique	0.83
Current Profile combined with error E_{zero} and error E_{pn}+E_{np} reduction techniques	0.84

Table 4.6: Overall accuracy of classification using different error reduction techniques

4.7 Related work

We have seen an overview of a number of works that are based on Twitter data in Chapter 2. Some of the relevant works include [SFD$^+$10], [SST$^+$09], [PP10], [JZSC09], which we discussed in Section 2.3.1. In this section, we present the works that involve classification of tweets with respect to an entity.

Many works based on entity identification and extraction, for example in [BM05, CKM09, KCMN08, YMA10b], usually make use of the rich context around the entity reference for deciding if the reference relates to the entity. However, in the current work, the tweets which contain the entity references usually have very little context, because of the size-restrictions of tweet messages. Our work addresses these issues, namely how to identify an entity in scenarios where there is very little context information.

The paper [TKW10] proposes a technique to retrieve photos of named entities with high precision, high recall and diversity. The innovation used is query expansion, and aggregate rankings of the query results. Query expansion is done by using the meta information available in the entity description. The query expansion technique is very relevant for our work, it could be used for better entity profile creation.

Bishop [Bis06] discusses various machine learning algorithms for supervised and unsupervised tasks. The task we are addressing in this work is generic learning, which can be seen as in between supervised and unsupervised learning. Yang *et al.* [YDT06] discuss generic learning algorithms for solving the problem of verification of unspecified person. The system learns generic distribution of faces, and intra-personal variations from the available training set, in order to infer the distribution of the unknown new subject, which is very related to the current task. We adapt techniques from [Bis06] and [CLY07] for the tweets classification task.

There are many ways to represent entities. In the Okkam [MBB$^+$10] project, which aimed to enable the Web of entities by offering an global entity identification service, an entity is internally represented as a set of attribute-value pairs, along with the meta information related to the evolution of entity. In DBpedia[13] and in Linked Data[14] the entities are usually represented using RDF models. These rich models are needed for allowing sophisticated querying and inferences. Since we use the entity representation for our classification algorithms, we resort to representing an entity simply as a bag of weighted keywords instead of the rich representations of entities.

In [PTPCR11] the authors address the problem of company identification in the micro-blogs by resorting to clustering techniques as a parallel approach to designing classifiers. They propose techniques

[13] `http://dbpedia.org/`

[14] `http://linkeddata.org/`

to improve the representation of a twitter message through term expansion, in a process to enrich the semantic similarity hidden behind the lexical structure.

Authors in [DFD11] look into similar problem in a different setting. They address the problem of filtering twitter messages for Social TV purposes. They are concerned if a tweet message is about some popular TV show (Lost, Survivor, Friends etc). Their approach, somewhat similar to ours, is of bootstrapping a model with smaller training set, developing a more sophisticated model using large dataset of unlabeled messages and further using domain specific features to obtain a final classifier. However, their focus was on developing a generic classifier that can be used on any unseen TV show in the training set.

We summarize the different classifiers proposed for the WePS-3 challenge task [AAG$^+$10].

The approach presented in [Kal10] uses data extracted from the company Web-site as surrogate training data. This data is used to create a initial model, which is then used to bootstrap a model from the Tweets. The model is iteratively refined with subset of tweets which were confidently classified by the model. The features used are the co-occurring words in each tweet and the relevance of them was calculated according to the Point-wise Mutual Information (PMI) value. Although it seems to be an interesting approach, the results shown provided a lot of scope for improvement. This system –even though it has low on overall accuracy– had decent F-score for relevant tweets, suggesting that a bootstrapping step can be very useful for company names with high ambiguity.

The authors in [CVSPO10] based their approach on linguistic aspects like recognizing named entities, extracting external information and making use of predefined rules. They use the well-known Name Entity Recognizer (NER) included in GATE (General Architecture for Text Engineering) for recognizing all the entities in their Tweets. They also use the Web page of the organization, Wikipedia and DBpedia to extract the company related information. Predefined rules are then applied to determine if a Twitter message belongs to an organization or not. The performance of the classifier varied across various companies. It is difficult to predict for what kind of companies this classifier performs well.

The research presented in [YMO$^+$10] proposes a two-phase system. In the first phase, they divide the organizations in the training set into 3 or 4 categories depending on the ratio of positive tweets to negative-tweets. In the second-phase, based on simple rules, the classification is done based on the category specific features extracted from the tweets. Their approach is based on the observation that the ratio of positive or negative (if the tweet is related to the organization or not) has a strong correlation with the types of organization names i.e. "organization-like names" have high percentages of tweets related to the company and when compared to "general-word-like names". Their system performance demonstrated high precision for positive examples and high recall for negative examples.

Another approach is described in [TB10], where the focus is on working with organization independent features and not relying on any external information sources. They trained the well-known J48 decision tree classifier using as features the company name, content value such as the presence of URLs, hash-tags or is-part-of-a-conversation (through re-tweeting, denoted in the messages with "RT"), content quality such as ratio of punctuation and capital characters and organizational context. This approach is quite interesting but heavily relies on the availability of training set. In our work we did not exploit the presence of hash-tags or re-tweeting behavior of users.

The basic profile classifier, discussed in Section 4.3, is based on the LSIR-EPFL classifier [YMA10a], which was the winner of WePS-3 evaluation challenge. The LSIR-EPFL classifier essentially makes use of different information sources on the Web to create an entity profile. We used these profiles for classifying the tweets. We further extended the basic techniques in [YMA11]. The current work is a long version of [YMA11], which gives further details on the work and introduces systematic performance analysis. The same dataset and company profiles were also used in an another line of research on designing quality-aware similarity functions for Web data, in [YMA12b].

4.8 TweetSpector: Entity-based retrieval of Tweets

Online Reputation Management (ORM) involves organizations monitoring the media, analyzing relevant content, finding what people say and feel about the organization entity, and if needed interact with the people. In this section we present *TweetSpector*, an application which can help companies and other entities to find relevant tweets concerning the entity.

As discussed earlier, we have seen that people readily express their opinions about the various products, companies, TV shows etc., on Twitter[15]. These tweet messages are thus a rich source of information that can be exploited to understand the sentiments about the concerned products or services. Retrieving the tweets related to given entities is however a challenging task as their names are often (deliberately) ambiguous, e.g. Apple, Blackberry, Friends, etc. Nevertheless, identifying the relevant entities is an essential first step to develop reliable sentiment analysis techniques that is not considered in existing systems, for example TweetFeel[16] and TwitterSentiment[17].

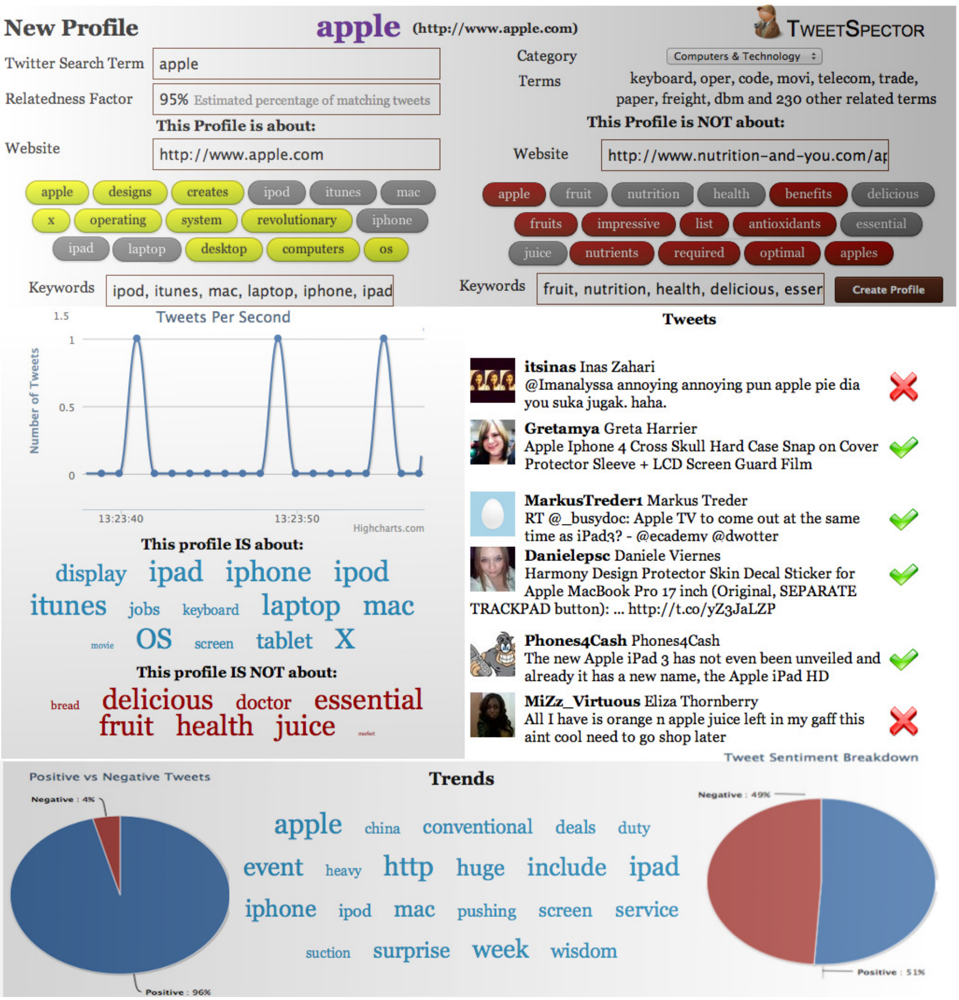
While there is a number of techniques for identifying named entities in unstructured text, they are often not directly applicable in this case, as tweet messages are very short (maximal 140 characters). Here we discuss *TweetSpector*, a tool that addresses this retrieval task and enables to link tweet messages to a given entity. Our retrieval methods rely on classification techniques that exploit our concise descriptions of entity-relevant information, also called entity profiles.

Figure 4.10 shows a number of features that are supported by *TweetSpector*, which are:

Entity Profile Creation: *TweetSpector* supports automatic profile creation, where we apply named-entity recognition, NLTK, wordnet and Web data extraction techniques to construct profiles for an entity, given a relevant Web-page. *TweetSpector* also enables manual profile construction, where users can construct arbitrary entity profiles, as well as manual and automatic updates for initially constructed profiles (thus the profiles are dynamic). The profiles can also be visualized using Word Clouds. The company entity profile creation and word cloud visualization can be seen in Figure 4.10 (top and center-left parts).

Realtime Tweet Classification: *TweetSpector* displays in real-time the classification results (see Figure 4.10, center-right part). For example, a stream of tweets is displayed and it is indicated whether or not the messages shall be related to the company Apple Inc.. The classification techniques are based on the number of techniques we presented in 4.4.

[15] `http://www.twitter.com`
[16] `http://www.tweetfeel.com`
[17] `http://twittersentiment.appspot.com`



New Profile **apple** (http://www.apple.com)

Twitter Search Term: apple

Relatedness Factor: 95% Estimated percentage of matching tweets

This Profile is about:

Website: http://www.apple.com

Keywords: ipod, itunes, mac, laptop, iphone, ipad

Category: Computers & Technology

Terms: keyboard, oper, code, movi, telecom, trade, paper, freight, dbm and 230 other related terms

This Profile is NOT about:

Website: http://www.nutrition-and-you.com/ap

Keywords: fruit, nutrition, health, delicious, esser

Tweets

itsinas Inas Zahari
@Imanalyssa annoying annoying pun apple pie dia you suka jugak. haha. ✗

Gretamya Greta Harrier
Apple Iphone 4 Cross Skull Hard Case Snap on Cover Protector Sleeve + LCD Screen Guard Film ✓

MarkusTreder1 Markus Treder
RT @_busydoc: Apple TV to come out at the same time as iPad3? - @ecademy @dwotter ✓

Danielepsc Daniele Viernes
Harmony Design Protector Skin Decal Sticker for Apple MacBook Pro 17 inch (Original, SEPARATE TRACKPAD button): ... http://t.co/yZ3JaLZP ✓

Phones4Cash Phones4Cash
The new Apple iPad 3 has not even been unveiled and already it has a new name, the Apple iPad HD ✓

MiZz_Virtuous Eliza Thornberry
All I have is orange n apple juice left in my gaff this aint cool need to go shop later ✗

Figure 4.10: TweetSpector: Various Features

User Feedback: The users can indicate whether the proposed classification is correct or not. This feedback is taken into account by the algorithms. The RIGHT and WRONG symbols shown in 4.10 (center-right part) can be toggled by the user through clicking those icons. *TweetSpector* can also take human input through crowdsourcing (through an interface to Amazon Mechanical Turk).

Dashboard: *TweetSpector* can display performance metrics and statistical information on a dashboard related to the entity. For example, one can observe trends and fraction of relevant tweets in Figure 4.10 (bottom).

Figure 4.11 shows the workflow involved in *TweetSpector* system. Let P_i be the current profile of the company at a time instant t_i and with a quality metric q_i. Say from time instant t_i to t_{i+1} we inspect and classify W number of tweets from the real time twitter stream using the current company profile P_i. We apply algorithm 4.1 over this W tweets with P_i as the starting profile, and say Δ_{i+1} is the knowledge of new keywords we gain from this window of tweets. We would combine P_i and Δ_{i+1} to obtain the new profile P_{i+1}, with quality q_{i+1}. For better performance from the profiles evolving over time, it is essential

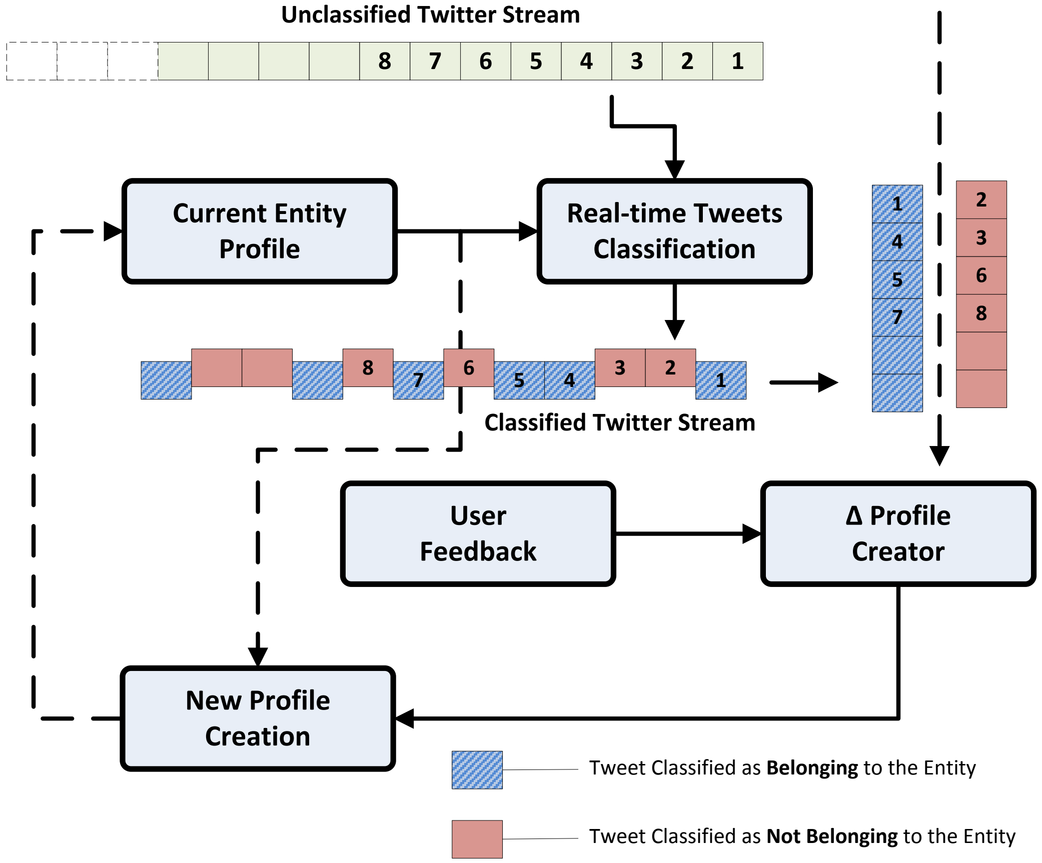

Unclassified Twitter Stream
8
7
6
5
4
3
2
1
Current Entity Profile
Real-time Tweets Classification
8
7
6
5
4
3
2
1
Classified Twitter Stream
1
4
5
7
2
3
6
8
User Feedback
Δ Profile Creator
New Profile Creation
Tweet Classified as **Belonging** to the Entity
Tweet Classified as **Not Belonging** to the Entity

Figure 4.11: TweetSpector Flowchart

that the quality of the profile should keep on improving i.e, $q_{i+1} \geq q_i \; \forall i$. Instead of automatically updating the current profile, our current version of *TweetSpector* relies on user feedback before updating the current profile. In our future work we would like to explore techniques of automatically updating the profile, and involve the user minimally. Our system also provides a way of visualizing the evolution of a company entity profile overtime.

4.9 Conclusion and future work

We studied how to classify Twitter messages containing a keyword, whether they are related to a given company, whose name coincides with the keyword. We proposed several techniques. First we presented a simple Naive Bayes classifier, which relies on automatically or semi-automatically constructed profiles. The company profiles contain two sets of keywords, which indicate whether a tweet containing this keyword is related to the company or not. We then extended this basic technique in two ways. First we developed a method, which takes estimations of the general ambiguity level of the problem into account. We have also introduced a technique that updates our company profiles actively from the twitter stream.

The main advantage of our technique is that it opens the possibility to estimate the accuracy of our classification decision. Indeed, we have exploited this possibility: we analyzed the sources of lower accuracies and we introduced methods to systematically address these problems. In this way we can minimize the uncertainty that is involved in the classification decision. We demonstrated how to localize the cases, where the human input is necessary, that is usually expensive to obtain.

In this way we can handle also the dynamic frequency changes in the use of words in the twitter language. Such changes arise naturally when a company temporarily receives media attention (e.g. if they launch a new product). Our experiments show systematic improvements as we extend our classifier with the described techniques. Though we demonstrated our techniques of entity-based classification on twitter messages, these techniques readily apply for other data sources like comments on social networks or blogs. Equally, one could apply the technique for other types of entities (for which we can obtain similar profiles) as well. In the end we also presented our prototype *TweetSpector*: entity-based classification of the tweets.

Part IV

Entity Profiling and Applications

Chapter 5

Social Networks based User Entity Profiles and Applications

You are what you share.

Charles Leadbeater, We Think: The Power Of Mass Creativity

Pervasive web and social networks are becoming part of everyone's life. Users through their activities on these networks are leaving traces of their expertise, interests and personalities. With the advances in Web mining and user modeling techniques it is possible to leverage the user social network activity history to extract the semantics of user-generated content. Entity Profiling is the problem of constructing a compact summary of an entity based on the content related to the entity. In this chapter, we explore various techniques for constructing user entity profiles based on the content they publish on social networks. We further show that one of the advantages of maintaining social network user profiles is to provide the context for better understanding of microposts. We propose and experimentally evaluate different approaches for entity disambiguation in social networks based on syntactic and semantic features on top of two different social networks: a general-interest network (i.e., Twitter) and a domain-specific network (i.e., StackOverflow). We demonstrate how disambiguation accuracy increases when considering enriched user profiles integrating content from both social networks. We also present *TripEneer* prototype that is based on user and location entity profiles. *TripEneer* is a user-based travel plan recommendation application.

5.1 Introduction

With the advent of Web 2.0, people being part of many social networks express themselves on various on-line platforms. A part of the users personality is latent among the different actions performed on social networks that they use. Given such user-generated data, it is possible to infer some components of user's personality and accordingly construct user profiles.

For example, an expert in map-reduce and cloud technologies would publish content more often about these technologies as compared to the average user. It could be through writing blog posts, or through microposts on Twitter, or through answering questions on Community QA (CQA) websites. In some cases, user generated content carries clues of user expertise and interests. Thus it becomes, in general, possible to infer expertise model from the user-generated content. Accurately constructing user entity profiles from their generated content is useful in many scenarios, such as:

Snapshot View : The user profiles we construct provide a summarized view of the user presence on on-line social networks.

Enhanced User Tagging on a Social Network : The user can be suggested with new tags (learned, e.g., from his Twitter network) which describe himself on a new social network (e.g., on StackOverflow[1]).

Enhanced Recommendation : Better recommendation engines can be built which can make recommendations based on the constructed user profile.

Information Filtering: The generated user profile can be used to filter relevant information from a stream of Web content based on the user interests.

In this chapter, we show a number of techniques of constructing user social profiles, we discuss their merits and demerits, and experimentally compare each of the techniques for constructing such profiles on the task of entity disambiguation. The different user profiling techniques we propose are:

Term Popularity: This method reports the top words of a user based on the observed frequencies of the different terms in the user generated content.

TF-IDF: In this method we consider those top words after sorting them based on their TF-IDF score.

Semantic: We make use of semantic techniques to extract concepts and categories from user-generated documents.

Topic Modeling: The top topics related to the user-generated content extracted using Latent Dirichlet allocation (LDA) topic modeling.

Labeled-LDA Topic Modeling: Summary of user-generated content in terms of labeled tags and words obtained by means of labelled LDA (LLDA) [RHNM09].

In this work we focus on using social network user profiles for effectively addressing the task of disambiguating entity mentions in social network content (i.e., understanding whether the mention of an entity like ‘apple’ refers to the fruit or to the company) [AAG$^+$10] by exploiting the content generated by users on other social networks. We explore how user profiles could be useful for extracting knowledge from data. Some examples of extracting knowledge from an unstructured data, like text documents, include named entity extraction [NS07], entity reference disambiguation [BT06], sentiment extraction [LZ12], linguistic tasks [OZHM13], etc. Various semantic and knowledge engineering techniques rely on the context for automatic meaning inference from a text [CKGS06]. Such techniques are successful for longer documents, as they provide enough context for the proposed tools. However, they can not be directly applied to short texts created within the social network platforms.

[1] `http://stackoverflow.com/`

Microposts are short texts posted by users on various social networks. Being short texts, microposts usually do not contain enough contextual information for making sense of them. While it would be difficult to develop new techniques that do not need such contextual information, we instead propose to use existing disambiguation techniques and rather to enhance the context of microposts by looking at user activity over other on-line social networks.

The proposed method for entity disambiguation in microposts is based on standard text classification using features extracted from the social network activities of the users. We experimentally compare the effectiveness of the proposed approach by disambiguating entity mentions in Tweets using as background information the user generated content on StackOverflow, a technical CQA system for computer programming topics.

Experimental results show that the classifiers built on top of enriched user profiles significantly outperform the classifiers built on top of the basic user profiles by at least 11%. The most effective approach is obtained using frequency-based and LLDA-based user profiles. By combining profiles constructed for a user over different social networks, it is possible to obtain a global social profile for the user which outperforms the other techniques in the tweet disambiguation task.

The rest of the chapter is organized as follows. Section 5.2 formally presents the user entity profile construction problem and the microposts disambiguation problem. Section 5.3 presents the overview of our approach. While Section 5.3.1 discusses a number of techniques for constructing user profiles, Section 5.3.2 discusses how to solve the microposts classification task. Section 5.4 provides a detailed description of the datasets and the experimental evaluation of the proposed user models. Section 5.5 summarizes the related work. Section 5.6 presents *TripEneer:* travel plan recommendation based on user social profiles. Finally, Section 5.7 concludes the paper.

5.2 Problem Statement

In this section we formulate the two tasks we are addressing in this chapter: the creation of a user profile from the user's social network content and the task of classifying a Twitter message based on its relatedness to a company entity.

Task 1: A user u_i publishes a set of micro-posts (ex: *tweets,comments*) on a social network. We group such microposts of a user together as a document $D_i = \{m_1, m_2, \dots, m_n\}$. We model the user profile U_i, of user u_i, as a bag of weighted set of keywords i.e. U_i = $Set\{wrd_k : wt_k\}$ with weights being normalized. These set of keywords could represent the topics or concepts that are most likely to occur in the user's microposts. We define $Corpus$ as the group of documents related to the various users of the system: $Corpus = \{D_1, D_2, \dots, D_m\}$. We define the topic extraction as a function f: D_i x $Corpus \Rightarrow U_i$. The techniques we considered are discussed in Section 5.3.1.

Task 2: Given a set of Twitter messages $\Gamma = \{T_1, \dots, T_n\}$ containing an ambiguous company name (e.g., *apple*, *orange*), we want to classify whether the message is related to a given company entity C or not. We say that the message T_k, created by user u_i, is related to the company C, $related(T_k, C)$, if and only if the Twitter message refers to the company. We also use the term that a tweet belongs to a company, by which we mean the same. We assume that some basic further information is available as input, such as the URL of the company $url(C)$ and the language of the Web page.

The tweet messages are modeled as a bag of words. Each tweet is preprocessed through following steps: we remove stop-words, emoticons, and Twitter specific stop-words (such as, for example, RT,@username); and we store a stemmed (using the *Porter stemmer*[2]) version of keywords (unigrams and bigrams). Formally we have: $T_k = Set\{wrd_j\}$.

The company entity C is modeled as a set of weighted keywords. The company entity: $C = Set\{wrd_k : wt_k\}$, with $wt_k \geq 0$ for positive evidence keywords (i.e. those words which suggest that the message should be related to the company) and $wt_k < 0$ for negative evidence keywords. We discuss the classification of tweet messages belonging to a company entity in Section 5.3.2.

5.3 System Overview

Users are typically present on several on-line social networks. They publish microposts on their Twitter stream, comments and posts on Facebook, address questions and answers on CQA sites like StackOverflow, express their interests through Facebook likes and Google +1s, etc. All the content users post, his activities on the web, and his social network interaction data can be tremendous value for automatically constructing a part of the user personality.

In this section, first we present a number of techniques for constructing user entity profiles. In the second part we address the problem of classifying a micropost (tweet) based on whether it is related to a company entity or not.

5.3.1 User Entity Profile Techniques

5.3.1.1 Frequency and TF-IDF based Topics

TF-IDF is often used in information retrieval and text mining for weighting document terms. A term is considered as important to a document if it appears more often in the document itself and tends to appear in fewer documents in the corpus. Term-frequency (TF) captures how often a particular word appears in a document, while inverse-document-frequency (IDF) captures how rare a particular term is in the document corpus.

$$tf(w_i, D) = \frac{freq(w_i, D)}{max\{freq(w_k, D); \text{for word } w_k \in D\}} \tag{5.1}$$

$$idf(w_i, D) = log\frac{|CorpusSize|}{\text{Number of docs containing the } w_i} \tag{5.2}$$

$$tf\text{-}idf(w_i, D) = tf(w_i, D) * idf(w_i, D) \tag{5.3}$$

A user entity profile U_i^{tf} is constructed based on the TF metric. We choose the top-K (with K ranging from 10 to 150) terms with the highest TF score (eqn. 5.1) to be present in the user entity profile. Similarly we construct another user entity profile U_i^{tfidf} based on the TF-IDF metric (eqn. 5.3). The top-K terms with the highest TF-IDF score are stored in this user profile.

The frequency based user profile U_i^{tf} is independent of the corpus, as it only depends on the current user-generated document. Such property allows a relatively efficient construction this profile. However, when a user tends to publish tweets related to various topics, for example: *technology*, *sports* and *politics*), and one of such topics is predominant, then the frequency based profile fails to capture the diversity in the different topics the user is writing about.

[2] http://tartarus.org/martin/PorterStemmer

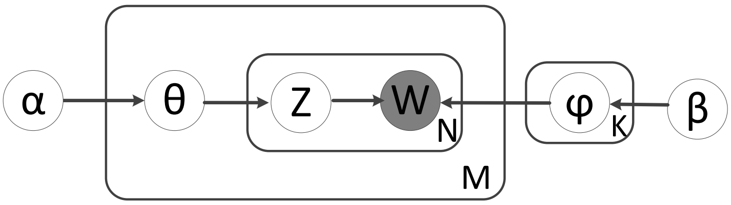


Figure 5.1: LDA Plate Model

5.3.1.2 Semantic-based Topics

Many semantic tools have been developed based on top of large document corpus like Wikipedia, News, Blogs. Example of such tools include: Alchemy[3], Calais[4], Textwise[5], etc. They are built using statistical natural language processing and machine learning techniques. These tools are inherently capable of extracting the semantic concepts, identifying named entities, assigning an hierarchical category label, etc. to a document based on its content.

To create a user profile we first group all the tweets of a user u_i into a single document D_i. We extract concepts and category labels from the document D_i using language modeling and neural networks[6]. The semantic-based user profile ($U_i^{semantic}$) contains the keywords representing such concepts and category labels. While such user profile has least number of keywords as compared by other approaches, it remains easy to understand and interpreted by a human.

5.3.1.3 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is an unsupervised learning algorithm that models each document in a corpus as a mixture of topics. The topics in turn are mixtures of words in the vocabulary. The latent variables of document to topics mixture distribution and topic to words mixture distribution are learned using the LDA technique.

Figure 5.1 shows the plate notation capturing the dependencies among different parameters of the model. α and β are Dirichlet priors on per-document topic distributions and per-topic word distributions. θ_i represents the topic distribution for a document D_i, while ϕ_k represents the word distribution for topic-k. w_{ij} and z_{ij} represent the word and the topic of i^{th} term in j^{th} document. K represents the number of topics and M represents the number of documents in the corpus. Among many variables, only the words w_{ij} are observed variables, while the remaining are latent/hidden variables. There are number of techniques for inferring the latent variables. In our current work we make use of collapsed Gibbs sampling [TMT] approach for inferring the latent variables of the corpus.

The output of the LDA learning process is topic-to-word distributions (ϕ_k) and document-to-topic distributions (θ_i). As for the frequency-based profiles, we extract top-K keywords (with K ranging from 10 to 150) after combining both these distributions θ_i and ϕ_k for a given user U_i, and group these keywords and term them as LDA-based user profile (U_i^{LDA}).

[3] http://www.alchemyapi.com/

[4] http://www.opencalais.com/

[5] http://www.textwise.com/

[6] See `http://www.textwise.com/api/documentation/introduction`

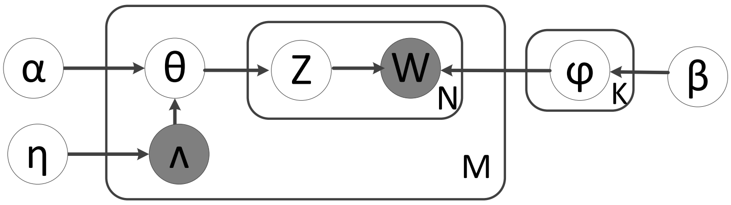


Figure 5.2: Labeled LDA Plate Model

U_i^{tf}	stack, overflow, google, app, software, developer, feature, generator, twitter, design, #stackoverflow, ...
U_i^{tfidf}	#annoyingsecurityquestions, #spoton, #shootingfishinabarrel, cinnabon, justintv, torah, #changetheratio, ...
$U_i^{semantic}$	stack exchange, computers, internet, protocols, arts, science fiction and fantasy, software, crafts, knitting and crochet, computers, open source, software
U_i^{lda}	startup, social, facebook, business, obama, romney, google, ...
U_i^{llda}	development, sharepoint, serial, compression, ms, graph, graphics, uml, azure, scriptaculous, ...

Table 5.1: Topic keywords extracted for a popular Twitter user and StackOverflow co-founder: Joel Spolsky (`@spolsky`)

5.3.1.4 Labeled Latent Dirichlet Allocation (LLDA)

LDA is an extremely popular model for summarizing a document corpus. However, it is not designed to handle multiple-labeled corpora, and it also suffers from the fact that inferred topics are not labeled thus needing a human to create topic interpretations. Labeled LDA ($LLDA$) [RHNM09] is a generative model for document collections that have labels assigned to each of the document. Topics extracted using LLDA are inherently labeled using the labels supplied with the documents. The topic-word distributions inferred during the learning process correspond to the label topics. Each label will have a multinomial distribution over the words found in the corpus.

Figure 5.2 shows the LLDA plate diagram. Most of the parameters are same as LDA parameters. Additionally we see variables (η and Λ) corresponding to the labels of the documents. In LLDA, the document is supervised to learn the topics corresponding to the attached labels. We use collapsed Gibbs sampling [TMT] for inferring the latent variables. Similar to LDA, we extract top-K keywords (with K ranging from 10 to 150) after combining the document-label distribution (θ_i) and label-word distribution (ϕ_k), and group them as LLDA-based user profile (U_i^{LLDA}).

5.3.2 Tweet Messages Classification

In this section, we address the problem of classification of a tweet message T_j that contains an ambiguous company name and posted by an user u_i, on whether it is related to a company entity C. As discussed in Section 5.2, we model the company entity C as a weighted set of keywords, where keywords act as positive or negative evidence depending on their weights. The tweet bag of words are compared against the company entity C bag of words. Depending on the amount of positive or negative keywords that are present in the tweet, it is classified as related to or not related to the company entity.

A tweet being a short message (maximum of 140 chars) would contain on average 10-15 words. As the tweet message contains very little context, the burden of better classification shifts to obtaining a better company entity C description. We construct an entity profile C following the findings of Yerva et al. in [YMA11] (also discussed in Chapter 4), where the authors identify multiple information sources to richly model the company entity profiles. They extract relevant keywords from the homepage[7] of the entity, keywords from the meta-data provided on the company web-pages, keywords from the glossary related to the category[8] of the company, keywords inferred using Google-set, or Wordnet services. They also rely on Wikipedia disambiguation pages for negative evidence keywords.

Moreover, the company entity profile C should not have too few words, resulting in less overlap with the tweet message keywords, therefore leading to random classification of tweets. On the contrary, the entity profile should not be too general, therefore avoiding many false positives during classification.

For our classification problem, we make use of Naive Bayes Classifier [Hec96, Lew98]. We assume the words appearing in a tweet independently contribute towards the evidence of whether the tweet belongs to the company, or not. We extend the model discussed in Section 4.2 of Chapter 4. Since we extend this model, we repeat it here for clarity and continuity.

For each tweet T_i = $set\{wrd_j^i\}$ we compute the conditional probabilities $P(C \mid T_i)$ and $P(\overline{C} \mid T_i)$ for deciding if a tweet belongs to a company C or not. We make use of Bayes theorem for computing these terms.

$$\begin{aligned} P(C \mid T_i) &= \frac{P(C) * P(T_i \mid C)}{P(T_i)} \\ &= \frac{P(C) * P(wrd_1^i, \ldots, wrd_n^i \mid C)}{P(T_i)} \\ &= K_1 \prod_{j=1}^{n} P(wrd_j^i \mid C) \end{aligned} \tag{5.4}$$

Similarly we have,

$$P(\overline{C} \mid T_i) = K_2 \prod_{j=1}^{n} P(wrd_j^i \mid \overline{C}) \tag{5.5}$$

where, $P(wrd_j \mid C)$ and $P(wrd_j \mid \overline{C})$ are the weights associated with the words wrd_j as described in the previous section. Depending on whether $P(C \mid T_i)$ is greater than $P(\overline{C} \mid T_i)$ or not, the Naive Bayes Classifier decides whether the tweet T_i is related to the given company or not, respectively.

[7] Ex: http://www.apple.com for *Apple* company entity

[8] Apple is a Computer Technology category company.

Another way of improving the tweet message classification is through enriching the context of the tweet. While there is no clear concise definition of context, the location and the time of the tweet message, the previous and next messages (neighborhood) of the current message, etc. could act as context of the message. In this work we use the user profile constructed using the different techniques to provide certain context to the message to be classified.

A user profile U_i corresponding to a user u_i, is modeled as a set of weighted keywords. We have already shown various techniques to construct such user profiles for the user generated content. When we combine the user context U_i with the tweet message T_j we get a new message, i.e., the tweet message in user context and we call it M_j. Even though there are many ways of combining the user profile U_i and tweet message T_j for obtaining M_j, we choose to focus on a simple union function. The resulting M_j will contain all the keywords found in U_i and T_j.

$$M_j = \cup\{T_j, U_i\} = Set\{\overbrace{w_1^j, \ldots, w_n^j}^{\text{Tweet words}}, \quad \underbrace{w_1^i, \ldots, w_m^i}_{\text{User Profile Keywords}} \quad \} \tag{5.6}$$

We again use Naive Bayes Classifier for classifying the context enhanced Twitter messages M_j. The conditional probabilities $P(C \mid M_j)$ and $P(\overline{C} \mid M_j)$, similar to eqns 5.4 and 5.5, decide if the original tweet T_j belongs to the company entity C or not.

$$P(C \mid M_i) = K_1 \overbrace{\prod_{k=1}^{n} P(w_k^j \mid C)}^{\text{Tweet Component}} \prod_{k=1}^{m} P(w_k^i \mid C) \tag{5.7}$$

$$P(\overline{C} \mid M_i) = K_2 \prod_{k=1}^{n} P(w_k^j \mid \overline{C}) \quad \overbrace{\prod_{k=1}^{m} P(w_k^i \mid \overline{C})}^{\text{User Profile Component}} \tag{5.8}$$

5.3.3 Cross Social Network User Profiles

More than just the features described above and their combination, we can exploit the fact that users participate on different social networks. Thus, we generate a *global social profile* that combines evidences from different social networks the user is involved in. This allows to take into account the diversity of content produced by users over different type of social networks (e.g., professional and leisure). By accounting the variety of content and meaning an entity can have for the user we aim at improving effectiveness of tweet classification. In the context of this work, we combine a general-interest social network (Twitter) with a domain specific one (StackOverflow) to build more diverse user profiles. Such enhanced profiles, obtained by merging the keyword lists from the best performing technique on each network, prove to be very useful when the company profile C is not extensive or noisy.

5.4 Experimental Evaluation

5.4.1 Data Description

We applied the user profile techniques explained in Section 5.3.1 on both a Twitter and a Stack Overflow[9] (SO) dataset. Stack Overflow is a website that features questions and answers on a wide range of topics in computer programming. Questions are tagged by the users (up to 5 tags)–at the moment of writing this chapter, the top-6 tags on the website are: C#, Java, PHP, JavaScript, JQuery and Android. Stack Overflow embeds also a simple but very effective reputation system that contributed to the spam-free user experience on the website. For instance, questions can be re-tagged only by users with a reputation score above 500 (i.e., users who have spent a fair amount of time contributing to the platform). For this reason, we consider StackOverflow tags as a set of "labels" carefully redacted by domain experts, hence a valid input to our LLDA user profiling technique; e.g., once a user writes a valid answer to a question tagged as "Scala", we can indeed infer that she has some expertise on the Scala programming language, hence defining a characteristic aspect of her profile.

Lacking a similar set of accurate labels for Twitter users, we employed LDA instead of LLDA. On the other hand, we applied TF, TF-IDF and Semantic on both datasets.

The evaluation dataset has been built with the following procedure:

- from the Stack Exchange Data Dump of August 2012[10], we identified 7772 users who reported their Twitter account in the Stack Overflow profile description
- for each of these users, we extracted all the data available in the StackOverflow XML dump: profile information, questions and answers, and tags (extracted both from the questions asked directly by the user and from the questions the user's answers referred to)
- we crawled Twitter (using the REST API) to obtain the latest tweets of the user (until Mar 12, 2013)

Users	6923
StackOverflow posts	592,021
Distinct SO tags	22,930
Tweets	4,894,944

Table 5.2: StackOverflow + Twitter dataset statistics

After cleaning the dataset (e.g., removing users with no activity, or with a protected Twitter account), we merged the information coming from both sources (Stack Overflow and Twitter) in a columnar database, to enable fast slicing and dicing of the user data.

It is worth to note that, due to the rate limiting in the Twitter REST API, we collected a maximum of 1000 tweets per user. On the other hand, the Stack Exchange Data Dump allowed us to process the whole history of the Q&A platform.

[9] Stack Overflow: http://stackoverflow.com/

[10] http://www.clearbits.net/creators/146-stack-exchange-data-dump

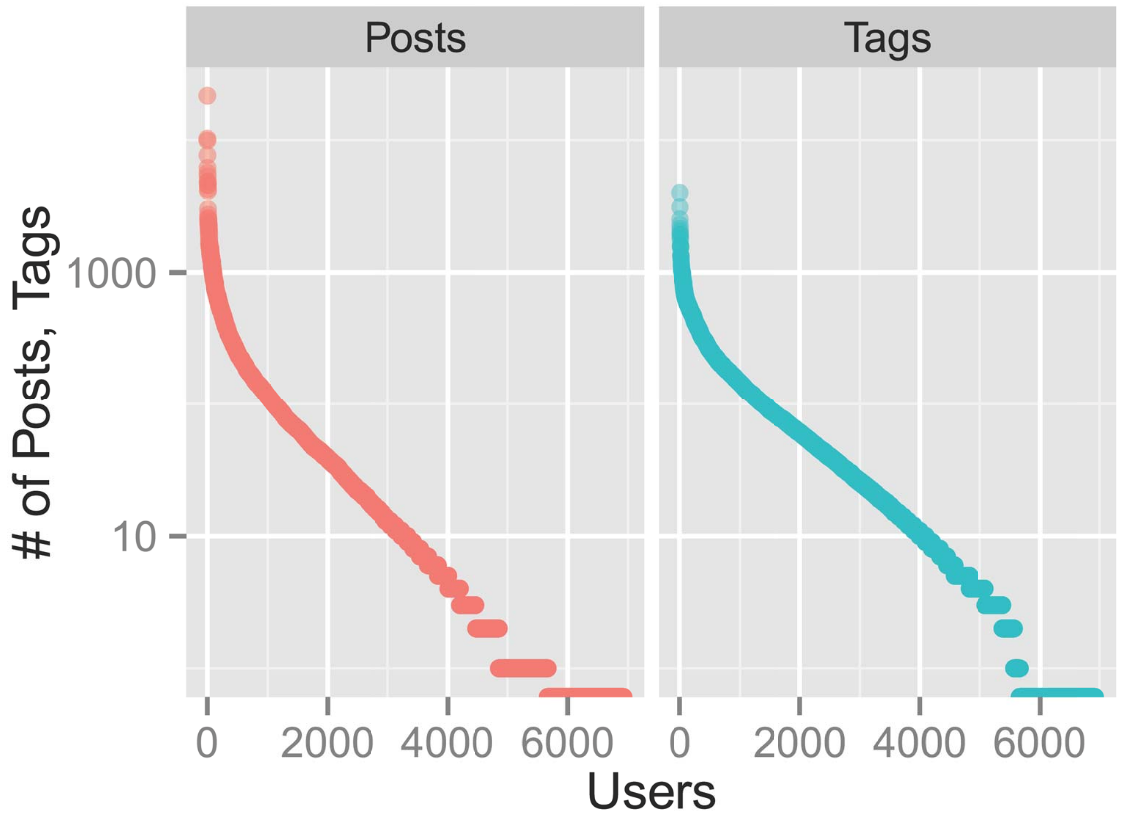


Figure 5.3: Power-law distribution of Stack Overflow Posts and Tags

Our sample of the Stack Overflow users' activities follows a power-law, as shown in Figure 5.3. Such distribution is very common in websites driven by user-generated content, confirming the validity of the approach followed to build our dataset.

5.4.2 User Profiles Construction

For each user, we extracted the text content of her tweets and StackOverflow content, and used it as an input for the 5 techniques explained in Section 5.3.1: TF, TF-IDF, LDA, LLDA, Semantic. While TF, TF-IDF and Semantic were applied on both social networks, we used LDA exclusively on Twitter, and LLDA exclusively on StackOverflow. TF, TF-IDF and Semantic return a ranked list of tokens, and for each we extracted the top-K results, with $K \in \{10, 25, 50, 75, 100, 125, 150\}$. LDA and LLDA, instead, required a more elaborated procedure. First, we computed the perplexity score for each model, varying the number of extracted topics. The perplexity score[11] measures how much the original corpus differs from one generated by the model trained on such corpus. Although it is expected that the perplexity score decreases with a higher number of topics, it does not give any guarantees on the quality and coherence of the topics. Furthermore, training a LDA or LLDA model does not scale gracefully with the number of topics (both in terms of CPU time and memory required). Given the results shown in Figure 5.4,

[11] http://en.wikipedia.org/wiki/Perplexity

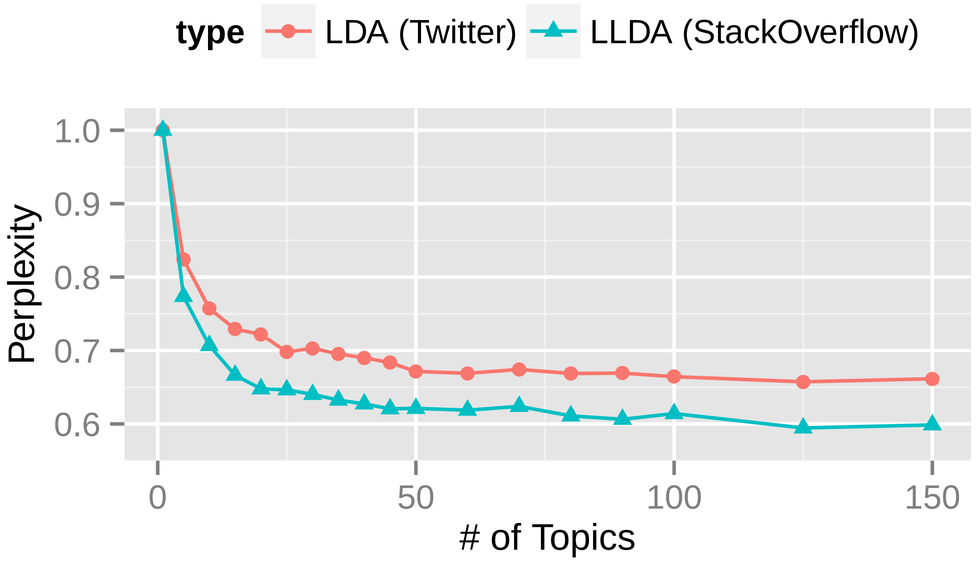


Figure 5.4: Perplexity on the Twitter and StackOverflow corpora (normalized to 1)

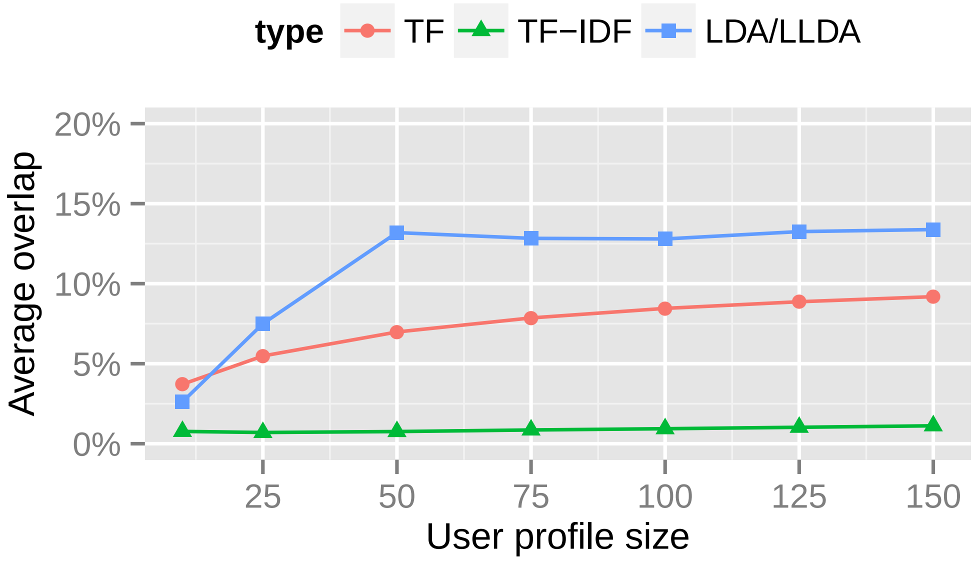


Figure 5.5: Average overlap between User Profiles extracted from Twitter and StackOverflow.

and after manual inspection of the generated topics, we opted to train our models with 50 topics, as it represented a good tradeoff between time spent by the training procedure and quality of the topics. Once the topics are generated, we run the inference process on the data of each single user, obtaining a ranked list of tokens which we sliced to extract the top-K keywords (with $K \in \{10, 25, 50, 75, 100, 125, 150\}$).

Figure 5.5 reports the average overlap between profiles extracted for a single user on both Twitter and StackOverflow. The overlap has been computed in the following way: for each user, we extract 2 top-K lists from both social networks, employing TF, TF-IDF or (respectively) LDA and LLDA. We then

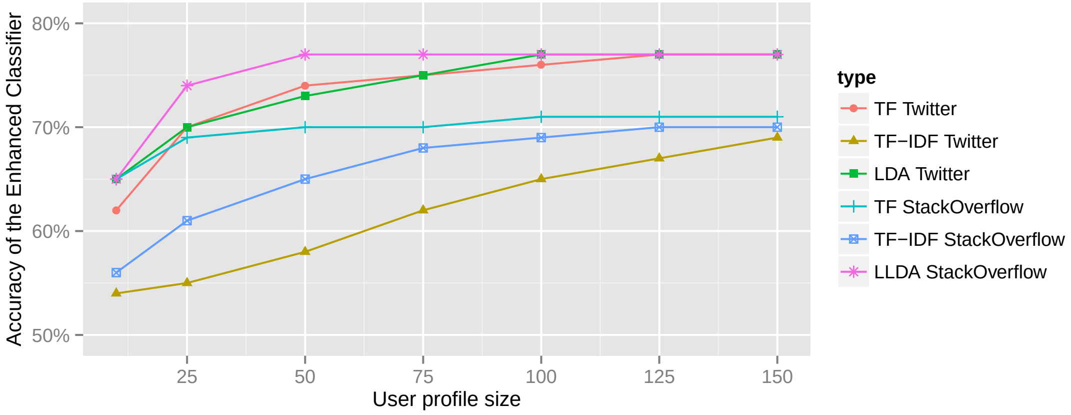


Figure 5.6: Enhanced Classifier performance with different User Profile techniques and sizes.

Dataset	apple	oracle	apache	subway	orange	seat
WePS3	0.83	0.78	0.47	0.45	0.05	0.02
SOTW	0.93	0.96	0.97	0.12	0.15	0.01

Table 5.3: Datasets Comparison: Percentage of tweets, containing the company keyword, that are related to the company Entity.

compare the two lists with the following similarity function:

$$Similarity = \frac{\sum_{i=1}^{K} get\_close\_match(topK\_TW[i], topK\_SO)}{K} \tag{5.9}$$

`get_close_match` is a function that returns 1 when it finds a fuzzy match between one of the tokens in the Twitter top-K and the StackOverflow top-K, 0 otherwise. The fuzzy matching is mostly based on the concept of string edit distance (i.e., Levenshtein distance), but the cutoff parameter has been set in such a way that almost only perfect matches would return a 1.

“Semantic” is not included in the Figure 5.5 because the technique we use does not return large sets of concepts, hence we cannot build Semantic profiles of different sizes. Similarly to LDA/LLDA though, the Semantic profiles are characterized by an average 11% overlap between Twitter and StackOverflow.

The relatively small overlap of the profiles built on different social networks is very valuable in our scenario, because it improves the diversity of the keywords used to disambiguate the tweets, as explained in the following section. It is also remarkable that, no matter the bias of our dataset towards high-tech oriented users, the profiles built on Twitter and StackOverflow show very different facets of the user.

5.4.3 Tweet Message Classification

WePS-3 dataset[12] contains tweets related to 100 company names, with an average of 500 tweets for each company name. The ground truth for each of this tweet is available in the dataset. However, we could not use WePS-3 for our experiments, because most of its tweets have not been posted by the users in our dataset. In fact, for comparison with our techniques, we need both the tweet message and the user who posted that message.

[12] http://nlp.uned.es/weps/weps-3/data

	BC	**Enhanced Classifiers (EC)**								
		Twitter				**StackOverFlow**				
Company	Basic	TF	TFIDF	Semantic	LDA	TF	TFIDF	Semantic	LLDA	Hybrid
apple	0.55	0.83	0.58	0.77	0.83	0.76	0.71	0.69	0.83	0.83
apache	0.5	0.52	0.51	0.51	0.52	0.52	0.51	0.51	0.53	0.53
oracle	0.55	0.77	0.64	0.55	0.78	0.7	0.66	0.58	0.78	0.78
orange	0.5	0.54	0.51	0.51	0.54	0.53	0.53	0.51	0.55	0.55
subway	0.54	0.94	0.68	0.82	0.95	0.83	0.78	0.57	0.95	0.95
seat	0.52	0.81	0.56	0.59	0.76	0.84	0.71	0.55	0.96	0.98
AVG	0.53	0.74*	0.58	0.63	0.73*	0.70*	0.65*	0.57	0.77*	0.77*
p-values		0.019	0.053	0.104	0.020	0.022	0.021	0.093	0.020	0.021

Table 5.4: Accuracy of the different classifiers: Basic Classifier and Enhanced Classifiers. Statistically significant improvement of EC over BC are indicated by * (t-test $p < 0.05$)).

	BC	**Enhanced Classifiers (EC)**								
		Twitter				**StackOverFlow**				
Company	Basic	TF	TFIDF	Semantic	LDA	TF	TFIDF	Semantic	LLDA	Hybrid
apple	84	0	75	16	0	20	35	40	0	0
apache	83	2	59	49	1	24	50	62	0	0
oracle	82	1	49	82	0	28	41	70	0	0
orange	87	13	76	66	11	21	33	75	0	0
subway	90	1	58	28	0	26	36	83	0	0
seat	95	34	86	80	45	29	56	89	4	0

Table 5.5: Percentage of non-overlapping Tweets with the Company Entity Profile. This percentage of tweets will be randomly decided by the classifiers. User profiles contain K=50 keywords.

From the 5 Million tweets we collected, we choose a subset of those tweets that contained at least one of the following set of six words: *apple*, *oracle*, *apache*, *subway*, *seat*, *orange*. The WePS-3 dataset contains 100 company names, with varying degree of ambiguity. We chose 6 company names as a representative sample of the entire dataset. Each of these 6 company names have multiple interpretations; e.g., the *apple* keyword could mean a fruit, the Apple company, New York city, etc. We are interested in classifying the tweet containing one of this keyword (for example: *subway*) with respect to its reference (or not) to the actual company (e.g., the *Subway* fast-food franchise).

For each of these 6 keywords, we manually annotated a total of 100 tweets, stating if they were related (or not) to their company entity. We refer to this dataset as the SOTW dataset. This manual annotation would act as ground truth for verifying the classification results of the two different approaches: one with

**	**BC**	**Enhanced Classifiers (EC)**								
		Twitter				**StackOverFlow**				
#CP	Basic	TF	TFIDF	Semantic	LDA	TF	TFIDF	Semantic	LLDA	Hybrid
0	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500
100	0.527	0.735* (0.019)	0.580 (0.053)	0.625 (0.104)	0.730* (0.020)	0.697* (0.022)	0.650* (0.021)	0.568 (0.093)	0.767* (0.020)	0.770* (0.021)
200	0.562	0.770* (0.020)	0.630 (0.071)	0.675 (0.079)	0.768* (0.020)	0.710* (0.019)	0.692* (0.021)	0.608 (0.079)	0.768* (0.020)	0.770* (0.020)
500	0.607	0.770* (0.039)	0.677 (0.084)	0.708 (0.078)	0.768* (0.038)	0.727* (0.037)	0.713* (0.035)	0.683 (0.078)	0.768* (0.038)	0.770* (0.039)
1000	0.633	0.770 (0.064)	0.702 (0.130)	0.713 (0.100)	0.768 (0.063)	0.730 (0.068)	0.718 (0.058)	0.693 (0.113)	0.768 (0.063)	0.770 (0.064)

Table 5.6: Average Accuracy Measure, along with p-values, for the different Classifiers w.r.t. varying quality of the Company Profiles. Statistically significant improvement of EC over BC are indicated by * (t-test $p < 0.05$)). **The first column $\#CP$ represents *Number of words in a Company Profile*

the classifier that takes the user profile into consideration, one with the classifier that does not.

Table 5.3 shows the percentage of tweets that belong to the company entity in the two different datasets: WePS-3 dataset and our dataset (SOTW). It is interesting to observe that the related percentages for tech company names (*apple*, *oracle* and *apache*) are higher in our dataset when compared to WePS-3. This is due to the fact that SOTW contains mostly tech-savvy users, while WePS3 is formed by a more general audience. Therefore, knowing the context in which a tweet was posted reduces the ambiguity in its interpretation.

Next, we compare the performances of the two classifiers: (1) Base Classifier (BC): the classifier which classifies tweets only based on the tweet keywords and the company entity keywords, (2) Enhanced Classifier (EC): the classifier that considers user profile keywords along with the tweet and company entity keywords for its classification task.

The performance of the classifiers depends on: a) the quality and size (K) of the user entity profile U_i; b) the size of the company entity profile C; and c) the percentage of tweets that contain overlapping words with the company profile words. We make use of the company entity profiles that were used in [YMA11, YMA12a]. As these company profiles were developed in the context of the WePS3 task [AAG$^+$10], we assume that their accuracies are bounded by the values in the first row of Table 5.3. Given the full-size company entity profile, we plot the accuracies of the classifiers by varying the number of words in the user profile, as shown in Figure 5.6. At $K = 50$, most of the user profiling techniques saturate the achievable accuracy of the classifier, suggesting that the user profile has already gathered a good candidate set of words for entity disambiguation. For this reason, we use $K = 50$ as the size of the user profile U_i, as it represents a good tradeoff between performance and computational cost.

We define the accuracy metric for the classifier as the percentage of tweets that are correctly classified. The performance of the classifier depends on the quality of the company entity profile C. Table 5.4 shows the accuracies of the different classifiers, for a fixed size company profile and a size of $K = 50$ of the user profile. Given a fixed company entity profile C, we see that the enhanced classifiers (EC) (that take user context into consideration) are outperforming the basic classifier (BC). The results in Table 5.4 and 5.5 clearly show that the user context helps the classifier in resolving the ambiguity involved in the company name. The percentage of tweets that do not overlap with the company profile in the test set represent the main cause of erroneous classifications.

In Table 5.5, we show the percentage of tweets in the dataset that do not have any overlapping keywords with the company profile C. The higher the number, the lower is the chance for a classifier to make accurate classifications. We see that the column-1 (basic) has the highest number of such tweets, while the remaining columns (that represent the tweets enhanced with user context) have a very low non-overlapping number of tweets. The Enhanced Classifiers are in a better position to classify the tweets more accurately, thus achieving our goal of “making sense of the microposts”.

Finally, we control the quality of the company profile C by varying its size, whose impact on classifier performance is shown in Table 5.6, along with the p-values (two tailed t-test). We observe that each of the Enhanced Classifiers (EC) is performing better than the Basic Classifier (BC), and this is true for all the size variations of the company profile. However, the percentage of improvement is statistically significant for lower sizes of the company profile. As it is relatively difficult to have an accurate company profile, based on our results we can benefit of the user social profiles especially when the company profile is noisy or too small.

Tables 5.4, 5.5 and 5.6 report also the results for a *Hybrid* technique, which merges the best techniques from multiple social networks to obtain a more diverse user profile. On our dataset, we observe that term frequency (TF) is best among the techniques applied on Twitter, and LLDA based is best among the techniques applied on StackOverflow. The resulting Hybrid user profile is then the top-25 for Twitter TF, combined with the top-25 LLDA for StackOverflow. Although the improvement of the Hybrid classifiers is not statistically significant on SOTW, we argue that its main advantage is represented by its reliable performance, regardless of the quality of the company profile. Our speculation is that, on a larger and more diverse dataset, the Hybrid classifier would systematically outperform the other Enhanced classifiers.

5.5 Related Work

Topic Modeling in Micro-blogging Platforms A number of recent works have explored the use of topic models in the Twitter domain for modeling Twitter messages and users [HD10], finding topical authorities [PC11, WLJH10, ZTL07], making recommendations [HBS10], and comparing it with other media [GAHY12, ZJW$^+$11]. We also focus our attention on works that have explored user modeling [AGHT11, GAHY12, HMOS12, AHK11] in micro-blogging platforms.

Works like, for example [LWH$^+$12, RCME11], have focused on adapting techniques and tools that were successful on text corpora to the recent vastly popular micro-blogging platforms. They adapted the named entity extraction (NER) techniques for the shorter and noisy micro-blog posts. The NER task is a critical step for the task of identifying the subset of tweets that are relevant to an entity which we tackle in this work.

Topic modeling of Twitter messages has been considered in [HD10], where models for three different tweet aggregation strategies have been considered: First, each Twitter message is considered as a document; second, all the tweets corresponding to a user are considered as being a single document; and finally, all tweets containing a particular term are put together in a one single document. These three strategies are referred to as MSG-Topic-Model, USR-Topic-Model and TERM-topic model. Each document D is considered to be sampled from a topic distribution (θ), and each topic has ϕ distribution over the words. The documents are generated based on the θ and ϕ distributions. One uses Gibbs Sampling to estimate the values of θ and ϕ. They show that the topics learned by the various schemes are different in quality. The topic models learned from aggregated messages of a user can lead to superior performance in classification problems. Based on their study, in our current work we grouped all the tweets corresponding to a user in to a single document and used it to infer the users' topics.

Several previous works [PC11, WLJH10, ZTL07, RDL10] have used topical modeling features on micro-blogging platforms for finding topic-based experts and authorities. The authors in their work on topical authorities in microblogs [PC11] propose various sets of features in order to find topic-based authoritative users. The set of features are based on how frequently users tweet, what percentage of their tweets are retweets, how often their tweets are retweeted, how often users are mentioned by other users, and how diverse or focused are the tweets to a particular topic. TwitterRank [WLJH10] proposes a ranking algorithm, an adaptation of PageRank algorithm, for finding topic-sensitive influential users. They make use of LDA on the twitter content for linking an user with certain set of topics, and use topic level similarity among users as feature of their ranking algorithm.

Expert finding in Social Network, combines personal local information with network information to find the experts on a topic. The approach proposed in [ZTL07] involves two steps: initialization and propagation. The initialization step forms an expert profile just based on the local information, and a propagation model is applied in the next step in which expert scores from one node are propagated to the neighboring nodes. Such approaches could be combined with the ones we propose in our work to improve the quality of both tweet disambiguation as well as of expert finding.

Most user interactions in Twitter are still primarily focused on the social graphs. Characterizing micro-blogs with topic models [RDL10] explores content analysis of Twitter feeds for addressing special information needs of the users. They apply LDA [BNJ03] and labeled LDA [RHNM09] for identifying the latent topics of Twitter messages. Using unsupervised LDA they assign latent topics into one of the four subcategories {*substance*, *social*, *status*, and *style*}. The partially supervised labeled LDA could assign labels (emoticons, hashtags, etc.) to the latent topics extracted from the Twitter feeds. We apply similar techniques for the problem of tweet disambiguation.

Some works, as in [ZJW$^+$11, GAHY12], have relied on topic modeling for comparing recent micro-blogging platforms and traditional news media platforms. In the paper [ZJW$^+$11], the authors do an empirical comparison of the Twitter content with that published on tradition media like the New York Times. Using standard LDA they infer topics from the news dataset, while they propose a Twitter-LDA model for extracting topics from Twitter data. This study shows how certain topics are popular on Twitter while some others are popular on news media. In [GAHY12] the authors extend their user modeling framework [AGHT11] for comparing the usage behavior on two popular micro-blogging platforms: Sina Weibo[13] and Twitter.

User Modeling over Micro-blogging platforms Web is gradually transforming itself as a users personal archive, where users not only find information but leave, share and archive information [LMB$^+$13]. Twitter being widely adopted, real time and representative of the users, despite being of noisy nature, is a great source for modeling a user [YMH$^+$]. User profiles were constructed in [SCS09, AGHT11, HMOS12] for better news and people-to-follow recommendations, dealing with information overload, understanding users' expertise and interests, etc. [SCS09] make use of entity profiles, that are sets of information extracted for each ambiguous person in the entire document, and features based on topic models to cluster documents –containing a person name– based on the actual person entity. Authors of [AGHT11] analyze user modeling on Twitter for personalized news recommendations. Their framework helps in creating user profiles that are based on extracted topics and entities from the tweet content, and show its superior performance compared to hash-tag based user profiles. They also consider temporal aspects of the user profile for better news recommendations.

The work [HMOS12] proposes techniques to construct multi-faceted user profiles for Twitter users, thereby helping one to navigate the complex domain-space represented by Twitter. Their model profiles users and their social networks using tags and labels from curated lists. In our future work, we plan to make use of the user maintained lists and the lists to which an user belongs in improving the quality of our constructed user profiles. [AHK11] work extracts professional interests from social web (Facebook, Twitter) profiles. Twittomender [HBS10] explores building of user profiles based on tweets which are grouped as users' own tweets, followers tweets and followees tweets. They make use of TF-IDF ranking

[13] `http://www.weibo.com`

technique in construction of the user profile, which they use for recommending other Twitter users to follow.

Micro-post Classification In [KML13, WC10] the authors present LDA transfer learning. Transfer Learning is the process of generic learning in one domain and applying the model in a different domain. In topic-bridged LDA ($tLDA$) a model is built from a variety of labeled and unlabeled documents, and they apply transfer learning for document classification task. One of our technique ($LLDA$) is based on transfer learning.

Several works [AAG$^+$10, YMA11, YMA12a] have addressed the problem of tweet classification in various contexts. For example, [YMA11, YMA12a] addresses the problem of Entity-based classification of tweets. Their techniques focus on accurately building the company entity profile, they also rely on *relatedness factor* metric of the company, and adapt active-learning for continuously improving their company entity profile. In our work, we focus on improving the classifiers performance by enriching the context of the tweet messages using the user social profiles.

5.6 TripEneer: User-based Travel Plan Recommendation Application

In the beginning of this chapter, we argued that accurately constructing user entity profiles could be useful for number of applications. In this section we present *TripEneer* as one such application that relies on the user entity profiles.

Current travel recommendation systems are helpful in addressing a traveler's information needs to certain extent, however, most of them fail to factor in the user in their recommendations. *TripEneer* proposes travel recommendations to a traveler by keeping the user preferences and constraints as first class citizens. We present an intuitive UI for helping users plan their travel trips quickly and easily. In this application we present various global and user-specific ranking models used for recommending travel destinations. Our preliminary evaluation showed that the users found the personalized recommendations, based on the user entity profile model, most useful.

5.6.1 Overview

Nowadays, spontaneous trips in popular European cities are made easy by the plethora of cheap means of transportation, visa relaxation policies and overall globalization. Because reaching the destination is quite easy, planning the activities there should be equally easy, but current online solutions do not cover this search space too well.

On the one hand, travelers can find generic landmarks by checking Wikipedia, Lonely Planet or via Google. These generic landmarks do not include any user preference and do not allow easy discovery of new landmarks. As it is common in power law distributions, where rich get even richer, most popular landmarks at a travel destination are promoted more often by these online systems, thus depriving the user of potential interesting landmarks that tend to occur in the long tail of power law distribution.

On the other hand with the advent of Web 2.0, it has become easier for a user to express himself on various social networking websites. It is possible to model the user and infer his preferences by taking his online activities into account. The recommendation systems, that consider the user online activities, can provide far more useful recommendations adapted to the user personality.

Additionally, once the user made up his mind about the activities to pursue, it is very difficult to obtain a map with the optimal route to visit the wanted locations. *TripEneer* is a prototype trying to solve these problems: *easily choosing the landmarks*, and *planning the trip path.* Focused on London, Rome, Boston and other popular cities, *TripEneer* gathered the landmarks available from Lonely Planet[14] and FourSquare[15] with their respective rankings. *TripEneer* proposes five ways of ranking these landmarks. Lonely Planet provided ranks are modeled using a power law distribution and a weighted average is computed with the rating provided from Foursquare. Another ranking is based on Foursquare signals: the sum of check-ins, tips and number of active users. In addition to these collaborative approaches, we rank landmarks by their proximity to locations visited by a user's Facebook friends. The Facebook profile information is used to compute a similarity value with each landmark and propose a fourth ranking solution. The last hybrid approach combines all the above using a user-specific weighted average mechanism.

Shi et al. use location data from user-uploaded photos and a collaborative filtering paradigm to recommend items favored by other users [YSL11]. We adapted this approach in presenting a ranked list based on what a user's Facebook friends visited, but the current work distinguishes itself by exploring other sources of location data and by recommending landmarks based on individual user profile instead of crowd-sourcing. Frankenplace [AM12] is an application for similarity-based place search that allows users to interactively find new places starting from features extracted from the travel blogs of existing places. On the contrary *TripEneer* proposes landmarks based on the user model.

Our *TripEneer* application is currently deployed here[16] and the users can start to interact with the system and explore a number of ranking schemes we proposed.

5.6.2 *TripEneer* Framework

Landmarks and Users are the main entities in our *TripEneer* framework. We use various data sources to richly model these two entities. Rich features of the Landmark entity are extracted from the data provided by Lonely Planet, FourSquare, Wiki-Travel and TravelBlog. The features include description, popularity, geo-location coordinates, events and images of the landmark. A User model, similar to the entity model [YMA12a], is developed from the features extracted from the user's Facebook, Flickr and Personal Blog profiles. Both the user and location entity profiles are built using the frequency-based topic model technique discussed in Section 5.3.1.

TripEneer proposes five different ranking models, based on the landmark and user features, for addressing the various users requirements. Figure 5.7 shows various tabs corresponding to the following ranking schemes.

Guides Ranking: Provides ranking based on the popularity of a landmark statistics accumulated by travel guides and by crowd-sourcing websites. The guide rank was modeled as a Zipf function. This value was averaged with the normalized crowd-sourcing rank.

Check-ins Ranking: Considers the normalized number of signals on Foursquare. These features indicate the activity around a landmark.

[14] http://www.lonelyplanet.com

[15] https://www.foursquare.com

[16] http://www.tripeneer.com

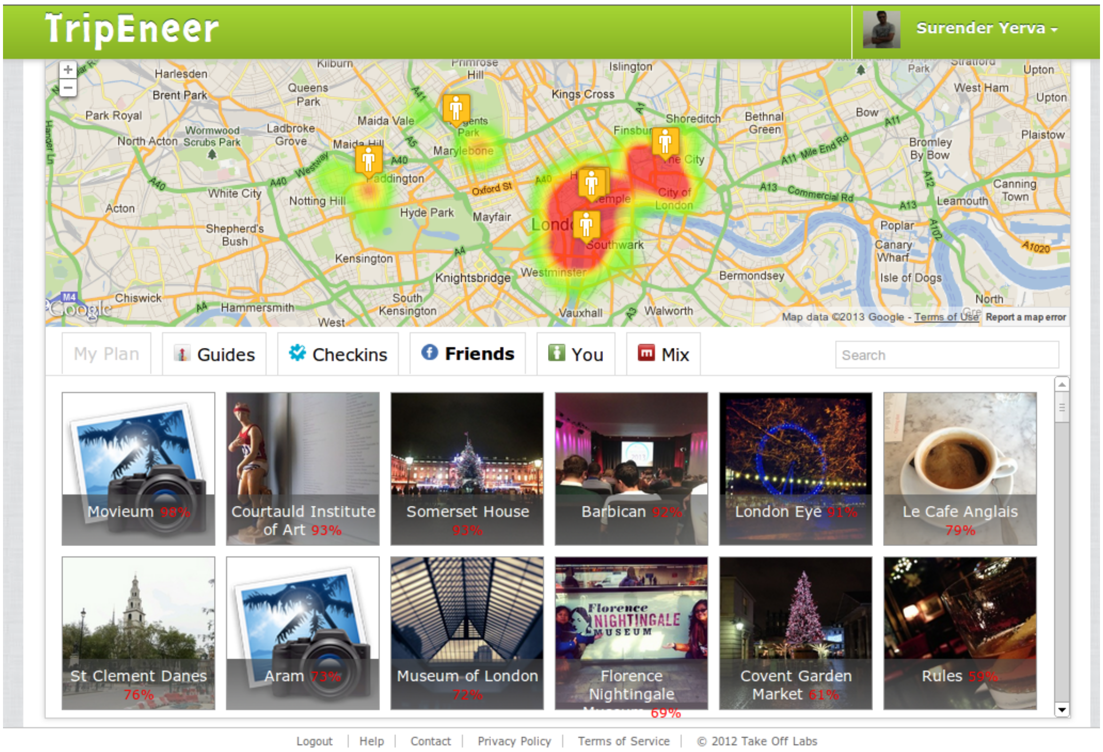


Figure 5.7: *TripEneer* Application: The various tabs showing different rankings. The heat-maps showing the popularity of various landmarks.

Friends-based Ranking: Ranking based on proximity to locations visited by a user's Facebook friends. The heat-map view provides a social context to the landmarks.

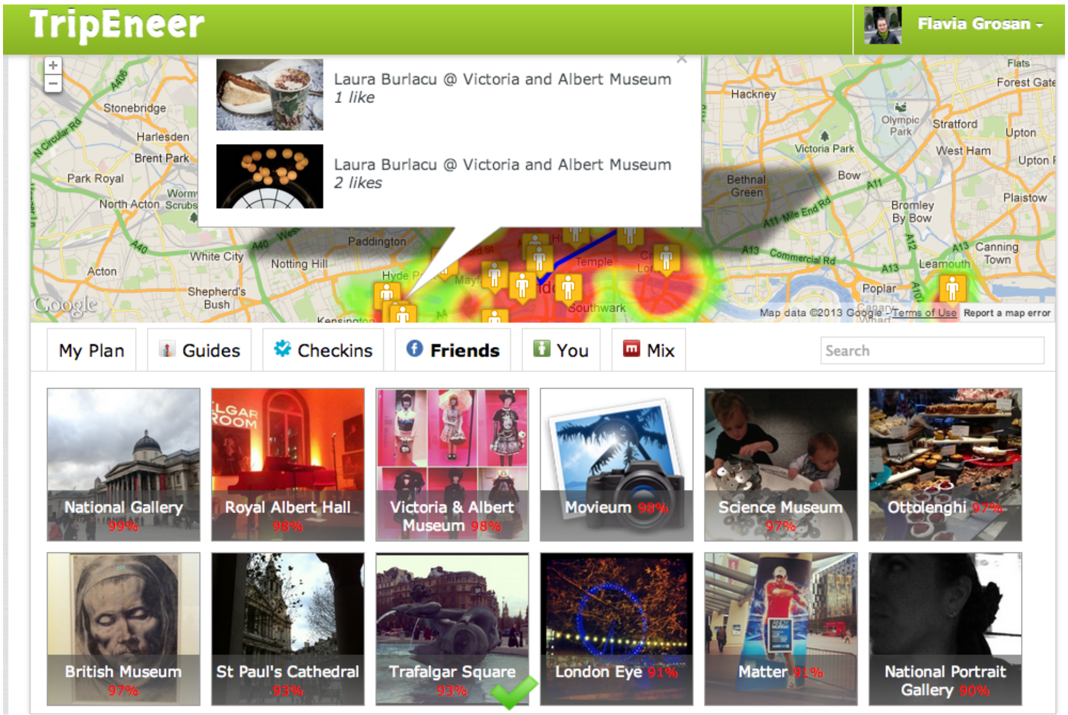
User-based Ranking: The user preference is modeled through Facebook profile information such as: pages liked by the user, about-me description and his posts. The landmarks are ranked using the distance-similarity function between the user model and the landmarks description.

Hybrid Ranking: The above four ranker values are averaged to obtain a combination of landmarks from all sections. The users can customize the weights for each score. In the future work, we plan to infer these values based on the users activity or through an interactive questionnaire to the user.

We have crawled many popular locations for the *TripEneer* application. It contains 10 locations and on average 370 landmarks per destination. The framework is developed on many scalable components and can be easily extended to many more destinations with little effort. In our preliminary evaluation we observed that personalized landmark recommendations were most useful to the user. The heat-maps corresponding to the landmarks visited by a user friends were informative and useful to the user. Additionally this ranking view (Figure 5.8) provided the user with list of friends, that visited this landmark, whom the user could contact for further information about that particular landmark, which we find as one of most useful feature of our application.

WorkFlow:

The user logins to *TripEneer* application using his Facebook credentials. *TripEneer* creates a user model based on the information extracted from the users Facebook profile. Next when the user chooses a travel



Laura Burlacu @ Victoria and Albert Museum
1 like

Laura Burlacu @ Victoria and Albert Museum
2 likes

Figure 5.8: *TripEneer* Application: The map showing the landmarks visited by users friends. The heat-map view provides a social context to the landmarks.

destination from the Dashboard (for example: *London*), the application provides landmark recommendations under different rankings tab. The user chooses different landmarks by exploring the different ranking tabs. The *MyPlan*-tab shows the set of landmarks chosen by the user. The map view provides a simple tour proposed by the framework.

5.7 Conclusions

Users in on-line social network generate content based on their interests and knowledge. They refer to entities which, in the given context are unambiguous for the other users who are consuming the content. However, to enable applications such as entity-centric search over social network content, we need to disambiguate the user generated content. In this work we presented a number of techniques for constructing profiles based on the content corresponding to an user entity, and evaluated their effectiveness for the tweet disambiguation task. Such user entity profiles present a summarized view of the user generated content across various social networks. In the second part of the chapter we have shown the importance of context in handling the tweet ambiguity: We used the user entity profiles to provide the missing context to the microposts, thus seeing an improved performance of the tweet classifier. Specifically, frequency-based features on Twitter and LLDA features on StackOverflow give user profiles that significantly improve effectiveness of disambiguation as compared to baseline approaches. Moreover, we have observed that the most reliable results are obtained by the combination of such best performing techniques to generate a global user profile that combines evidences from different social networks the user is involved in. In the current work we focused only on the user generated content, however, in future work we want to consider other information like the users social connections and their activities on the social networks for constructing better user profiles.

In the end we presented TripEneer, a personalized tour planning application. When a user is planning to visit a certain tourist location, the *TripEneer* application helps recommending the landmarks specific to his user profile, along with the general recommendations from Lonely Planet, Wiki Travel, etc. Users can view the landmarks both in the classical travel guide way, or can discover new places which match their preference. The users of the system liked the personalized rankings provided by the system. The friends-based ranking helped the users to readily identify which of their friends have visited these landmarks and can be contacted for further information.

Chapter 6

Social and Sensor Data Fusion in the Cloud

Torture the data, and it will confess to anything.

Ronald Coase, Economics, Nobel Prize Laureate

After seeing the profiling of an user entity, we shift our focus to profiling a location entity. We have already seen *TripEneer* (Section 5.6), which made use of location entity profile along with user entity profile for travel-plan recommendation to the user. Constructing entity profiles involves resolving all entity mentions corresponding to an entity, and then summarizing all integrated information into the entity profile. In this chapter we focus on fusion of social and sensor data corresponding to a location entity.

As mobile cloud computing facilitates a wide spectrum of smart applications, the need for fusing various types of data available in the cloud grows rapidly. In particular, social and sensor data lie at the core in such applications, but typically processed separately. Here in this work, we explore the potential of fusing social and sensor data related to a location entity in the cloud, presenting a practice—a travel recommendation system that offers the predicted mood information of people on where and when users wish to travel. The system is built upon a conceptual framework that allows to blend the heterogeneous social and sensor data for integrated analysis, extracting weather-dependent people's mood information from Twitter and meteorological sensor data streams. In order to handle massively streaming data, the system employs various cloud-serving systems, such as Hadoop, HBase, and GSN. Using this scalable system, we performed heavy ETL as well as filtering jobs, resulting in 12 million tweets over four months. We then derived a rich set of interesting findings through the data fusion, proving that our approach is effective and scalable, which can serve as an important basis in fusing social and sensor data in the cloud.

6.1 Introduction

Mobile phones increasingly become multi-sensor devices, accumulating large volumes of data related to our daily lives. At the same time, mobile phones are also serving as a major channel for recording people's activities at social-networking services in the Internet. These trends obviously raise the potential of collaboratively analyzing sensor and social data in mobile cloud computing—where applications running in the cloud are accessed from thin mobile clients, providing virtually unlimited processing power, and promising cross-device platform compatibility.

The two popular data types, social and sensor data, are in fact mutually compensatory in various data processing and analysis. Participatory sensing, for instance, enables to collect people-sensed data via social network services (e.g., Twitter) over the areas where physical sensors are unavailable. Simultaneously, sensor data is capable of offering precise context information, leading to effective analysis of social data. Obviously, the potential of blending social and sensor data is high; nevertheless, they are typically processed separately in mobile cloud applications, and the potential has not been investigated sufficiently.

In this chapter, we explore the possibility of fusing social and sensor data in the cloud, while dealing with massive data streams. To this end, we present a travel recommendation system as a practice of the fusion, which offers the information of people's moods regarding the predicted weather on where and when users wish to travel. The recommendation system gears various components towards effective, large-scale social and sensor data fusion. We summarize the salient features of the system in the sequel.

- First, we propose a conceptual framework that enables to integrate and analyze the heterogeneous social and sensor data. Specifically, the framework first transforms tweets into data points in a *mood space* which consists of 12 subspaces, each of which corresponds to a mood (e.g., happy). We then derive the probability of each mood in the mood space from a large number of tweet data points accumulated over time. The system computes and maintains the mood probability information separately according to day (e.g., Monday), place (e.g., London), and weather (e.g., sunny), which are the major dimensions in query processing.

- Second, we present a scalable fusion system that implements the conceptual framework, extracting the weather-dependent mood information from real-time Twitter and meteorological sensor data. Our travel recommendation system is established upon a combination of several well-known systems typically used for large-scale data store and analysis in the cloud, such as Hadoop [Whi09], HBase [HBa], and GSN [AHS06]. This allows us to perform ETL jobs as well as analytic processing over massively streaming data.

- Third, we offer in-depth analysis of our data-fusion approach on comprehensive experimental results, obtained from using 12 million tweets as well as meteorological sensor readings collected over four months. The results demonstrate various interesting findings, including the degree of happiness according to a particular weather type, day, and location. Furthermore, we statistically prove that our mood estimation based on the fusion is effective and accurate.

We believe that the approach proposed in this work can set a firm yard-stone in scalable social and sensor data fusion, serving as an important foundation in further studies towards mobile cloud computing.

The rest of the chapter is organized as follows. Section 6.2 summarizes the related work. Section 6.3 describes in detail the theoretical framework for fusing social and sensor data, while Section 6.4 presents the technical details as well as data collections used in our travel recommendation system. Section 6.5 offers experimental analysis on the data fusion, followed by the conclusions in Section 6.6.

6.2 Related Work

Social-network services facilitate users to share their ideas, opinions, pictures, videos, news, and other various forms of contents in the Web. Such social data typically contains highly valuable information, aiding a wide range of applications; for example, allowing social scientists to understand human behaviors, companies to figure out their customers' preferences, and news agencies to identify significant news etc. Previously, it was difficult to obtain the rich set of social information, or required large amounts of laborious human efforts like conducting surveys, interacting with the users. With the advent of Web 2.0, all this information is readily available, leading to a variety of interesting research directions. In this section, we summarize three research lines which are closely related to this study.

6.2.1 Mood analysis on tweets

One popular research line on social data is to extract and analyze mood information from Twitter messages [MBB$^+$11, BMZ10, Pul, TBP11, PP10]. In [MBB$^+$11] micro-blogs are used for mood analysis, where they present a method for associating mood to certain events. Their techniques help in summarizing huge volumes of tweets w.r.t. the events. The TwitInfo system proposed by the authors, allows users to browse a large collection of tweets using a timeline-based display that highlights peaks of high tweet activity corresponding to the events. Similarly, the authors of *Pulse of Nation* [Pul] by extracting sentiment information from Twitter messages are able to track the national mood. This study analyzed over 300 million tweets corresponding to the US region over a period of 3 years . They present the moods across the country using different cartograms; and observe the variation of nation's mood over 24-hour period of a day and the days of a week.

Another study [BMZ10] tries to predict the impact of public mood expressed in Twitter messages on the stock market; they do it by investigating the correlation of moods inferred from large-scale twitter feeds with the Dow Jones Industrial Average. They make use of mood tracking tools, namely, OpinionFinder (that measures positive vs. negative mood) and Google-Profile of Mood States (GoPMS) that measures mood in terms of 6 dimensions.

The authors of [TBP11] analyze Twitter messages in order to study why certain events resonate well with the population. They assess whether surges of interest in Twitter are associated with heightened emotions, by checking if the average sentiment strength of popular Twitter events is higher than the Twitter average, or by assessing whether an important event within a broad topic is associated with increased sentiment strength.

In [PP10], Twitter data is used as corpus for sentiment analysis and opinion mining, where Twitter becomes a media in which people readily express their opinion. Specifically, the Twitter data is served for training their sentiment classifier, which classifies tweets as expressing positive, negative or neutral sentiment.

6.2.2 Social sensing

Given the importance of sensor networks in our everyday activities, some studies [NSV11, RMZ$^+$11] went ahead and consider the people participating in micro-blogs or social networks as social sensors providing the rich social context, which are hard to infer using physical sensors. For instance, the work in [SOM10] monitors the flows of Twitter messages for quickly detecting an earthquake that occurs in an area where seismic sensors are unavailable. Another study [WYLL11] mines the Twitter messages to identify relevant events to given monitoring conditions. Yerva et al. [YMA11] also identify the tweets relevant to ambiguous company entities for its advertising strategies.

6.2.3 Social data fusion

CitizenSensing [NSV11] gives a broad overview of the challenges involved in making sense of citizen sensing, which is becoming rampant with ubiquitousness of the mobiles, sensing devices etc. The study introduces the paradigm of Citizen Sensing, enabled by Mobile sensing and Human Computing – humans acting as citizens on the ubiquitous Web, acting as sensors and sharing their observations and view through Web 2.0. Likewise, SocialFusion [BGX$^+$10] proposes the use of sensor networks to enable context-aware social applications, analyzing the data generated by the users of the applications. In addition, SocialSensors [RMZ$^+$11] describes the need for fusing social data with pervasive sensors for better services.

The authors in [LOIP10] present heuristic methods for data fusion that combine the user's personal calendar with his social network posts, in order to produce a real-time multi-sensor interpretation of the real-world events. Their study shows that the calendar can be significantly improved as a sensor and indexer of real-world events through data fusion.

6.3 The Fusion Framework

This section describes three major components of the theoretical framework in our data fusion approach, which are *fusion base, data points*, and *mood probabilities*.

6.3.1 Fusion Base: Mood Space

A key goal of this study is to establish a data-fusion approach that collaboratively analyzes both social and sensor data. In particular, we aim to extract people's mood information from social (Twitter) feeds associated with sensor (weather) data. To this end, we propose a data space, called *mood space*, which serves as a conceptual base-ground where social and sensor data can be mapped.

More specifically, we represent the mood of a word (e.g., appearing in a tweet) using the ANEW [BL99] list, which describes a set of major words frequently appeared in people's conversations as numerical scores. In ANEW, each of such words is scored in three dimensions: *valence*, *arousal* and *dominance*, where the value in each dimension ranges from 1 to 9. Valence is defined by its two poles negative/bad and positive/good, whereas the arousal dimension spans between the two poles sleepy/calm for very low arousal and aroused/excited for very high arousal. Valence and arousal have proven to be the two main dimensions, accounting for most of the variance observed. An additional dimension called dominance is proposed to differentiate subtle emotions like fear and anger (which have similar valence and arousal values).

In this study, we consider the mood space to be defined by valence and arousal metrics, illustrated as Figure 6.1. The two dimensional plane VxA:[1,9]x[1,9] is divided into 12 regions, each region maps to a certain mood. For example, a high value of valence and another high value of arousal indicates someone is happy, labeled as "happy" in the figure. Similarly, a lower value for valence and a high value for arousal maps to the mood of "annoying/rage".

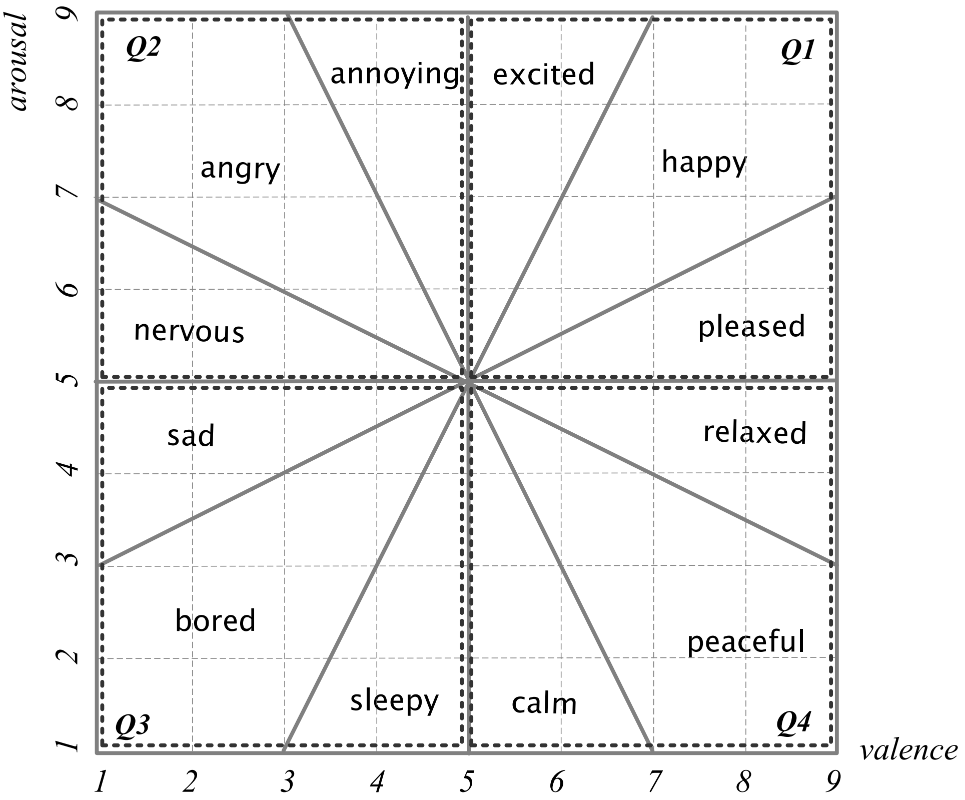


Figure 6.1: Illustration of the mood space.

6.3.2 Data Points: Tweet Mapping

Given a tweet message, the next step in our fusion approach is to associate a mood label with the message, by computing the valence and arousal scores of the tweet. Specifically, the tweet is decomposed into words, each of which would have a valence and arousal score, then we resort to Naive Bayes setting in order to compute the tweets' overall valence and arousal score which would become a data point in the mood space.

Formally, consider a set of moods M consisting of the 12 moods in the mood space. Given a tweet set T = $SET\{T_i\}$, for each tweet $T_i \in T$, first we try to infer the mood expressed by the tweet by computing the conditional probabilities P($M_k|T_i$) for all $M_k \in M$.

For computing the conditional probabilities P($M_k|T_i$), we resort to Naive Bayes setting. We consider a Tweet T_i as a bag of words, T_i = $set\{wrd^i_j\}$, and we assume each word expresses certain mood (the words which do not express mood will make zero contribution to the final mood). We assume each word independently contributes to the overall mood of the Twitter message.

$$\begin{aligned} P(M_k \mid T_i) &= \frac{P(M_k) * P(T_i \mid M_k)}{P(T_i)} \\ &= \frac{P(M_k) * P(wrd^i_1, \ldots, wrd^i_n \mid M_k)}{P(T_i)} \\ &= C_k \prod_{j=1}^{n} P(wrd^i_j \mid M_k) \end{aligned} \tag{6.1}$$

Each of the terms in the above product, P($wrd^i_j \mid M_k$), can be interpreted as the amount of contribution a particular word makes towards a mood M_k, which can be learnt based on the training set or one could readily use the weights provided by prior studies like the one in creating the ANEW list [BL99]. Along with the term weights, we also compute the constant C_k based on the training set. Depending for which mood M_k, the term $P(M_k|T_i)$ is largest, we classify the tweet T_i as expressing that mood.

For example, consider the following tweet T0 : *"Weather here is seasonal, warmish, some rain and sun, green and beautiful"*. This tweet is composed of 12 words, in which four of them are listed in the ANEW set of words. For these four words, we obtain the valence scores of (rain= 5.08; sun=7.55; green=6.18; beautiful= 7.60) and arousal scores of (rain= 3.65; sun=5.04; green=4.28; beautiful= 6.17) by looking up the ANEW list. Finally, applying the above procedure, we get the overall tweet valence and arousal scores as (6.60,4.78) which forms a data point in our mood space and gets a "relaxed" mood label.

6.3.3 Mood Probabilities

Given a set of data pointed in the mood space, derived from raw tweets, we next explain how the fusion framework computes a set of mood probabilities, according to day, location, and weather.

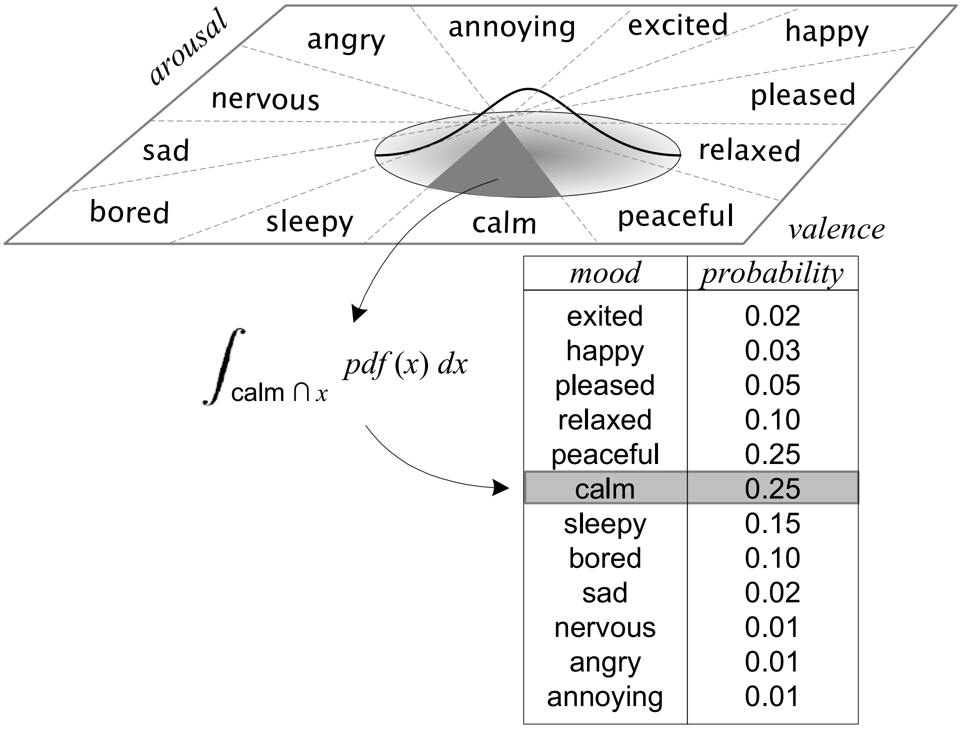
We know that each Tweet T_i carries the information about the location L, the time stamp t and the weather label W. Thanks to the analysis explained above, now the tweet also carries the mood M_i information. Now for each tweet $T_i \in T$ we have a record R : (T_i,L,t,W_j,M_i). Essentially each tweet now maps as a point in the 2D mood space. The complete set of twitter data maps onto the 2D mood space as a distribution of points. For easier querying our next goal is to summarize the distribution of points on the social metric space.

Once we have all the tweet records R's, one can summarize the mood-weather information using p_{ijk} probabilities. The p_{ijk} represents the probability of witnessing mood M_i when the weather is W_j and the day is $D_k \in \{$Monday,. . .,Sunday$\}$ i.e., the conditional probability $P(M_i \mid W_j, D_k)$. One can consider different models for computing this p_{ijk} probabilities, ranging from simple model which summarizes all the events so far ignoring the temporal aspects like time, weekday etc., to far more sophisticated models which give more importance to the recent events.

According to the simple model, we group all the tweet records corresponding to a particular location L. We observe different weather labels W_j, mood labels M_i and day labels D_k information associated with each of these tweets. Next we compute, p_{ijk} (shown in eqn. 6.2), as the fraction of tweets expressing certain mood M_i for a particular weather label W_j and the day D_k.

$$p_{ijk} = \frac{\#(\text{tweets with } M_i,\, W_j \text{ and } D_k)}{\sum\limits_{a=1}^{12} \#(\text{tweets with } M_a,\, W_j \text{ and } D_k)} \tag{6.2}$$

If we plot all the tweets satisfying the conditions of having certain weather label W_j and are on certain day D_k, as points on the mood-space, one would expect to see distribution of points over each mood space similar to the one shown in Figure 6.2. For a particular day and weather label, the fusion process helps us to obtain the probability distribution over the mood spaces. These probability distributions will be summarized for all days and weather labels combinations and will be used as source of useful input to the travel recommendation system.



mood	*probability*
exited	0.02
happy	0.03
pleased	0.05
relaxed	0.10
peaceful	0.25
calm	0.25
sleepy	0.15
bored	0.10
sad	0.02
nervous	0.01
angry	0.01
annoying	0.01

Figure 6.2: An example of mood probability computation.

6.4 The Travel Recommendation System

This section introduces a travel recommender system as a practice of the data-fusion framework described in the previous section. We first offer an overview of how the system works, and then describe each component of the system, as well as data processing.

6.4.1 Overview

The intuition behind the system development is to show that the information derived from various, real-time data fusion can enrich recommendations, compared with using solely static, limited-scope reviews posted by experts or other consumers.

In our recommendation system, users provide their travel intentions (place and approximate date of travel), and then the system provides the information of how enjoyable the place would be on the day for travel, in addition to the typical information offered by ordinary travel recommender systems. This recommendation process is comprised of the following steps:

1. A user first offers the details for travel to the system, e.g., going to London next Friday.

2. The system obtains the information of predicted weather on London next Friday, from a real-time weather prediction service (e.g., WeatherUnderground).

3. The system looks up the mood information of people associated London and Friday, which is continuously mined and updated from raw social and sensor data.

4. The system offers the information of how enjoyable the trip to London on next Friday would be, according to the mood probability estimation.

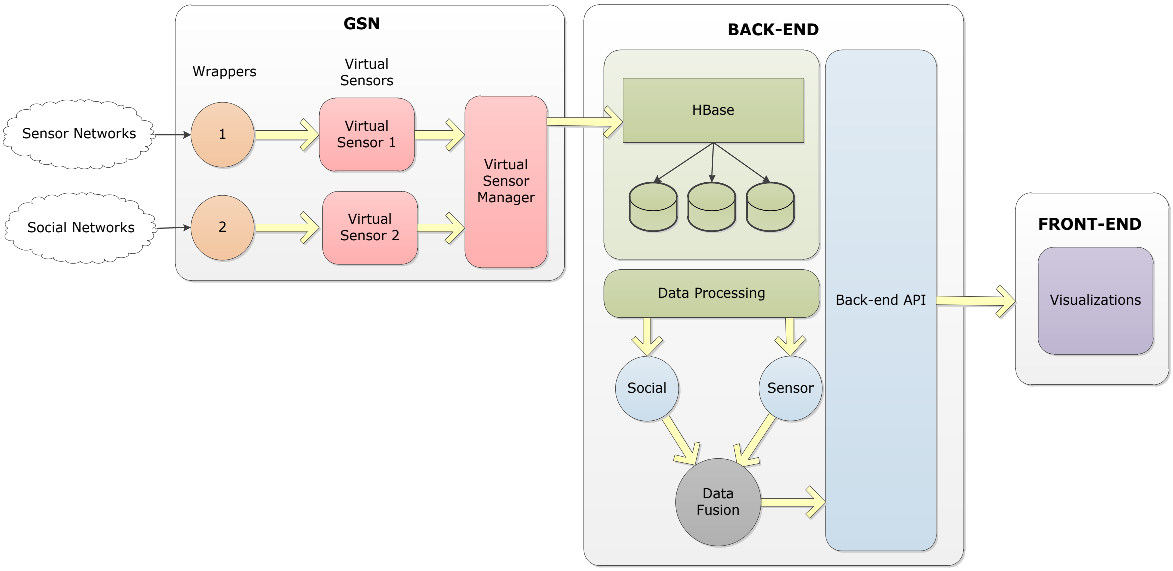


Figure 6.3: Architecture of the fusion system.

Note that our fusion system is flexible to blend other data sources with the social and sensor data, in order to make the recommendation more meaningful. For example, taking into account the events (e.g., death of a famous person, terrorism) occurring in London would be able to enrich the quality of recommendation. We believe that such an additional data source can be easily fused in the framework of the recommendation system.

6.4.2 System Architecture

In order to store and process massively streaming social and sensor data in the cloud, we propose a system established upon a combination of state-of-the-art cloud systems, including Hadoop [Whi09], HBase [HBa], and GSN (Global Sensor Network) [AHS06]. Figure 6.3 shows an overview of the system, which consists of three primary components. In the sequel, we describe in detail each of the components.

- *GSN* is a stream processing engine that supports a flexible integration of data streams. It has been used in a wide range of domains due to its flexibility for distributed querying, filtering, and simple configuration. In our travel recommendation system, GSN serves as a wrapper that receives streaming social as well as sensor data from twitter and weather data sources. GSN provides means to control the rate of data streams, and also allows us to parse and filter incoming data on the fly, before the data are stored in the back-end.

- *Back-End* contains both Hadoop and HBase, serving as a storage-and-computing platform. Hadoop (MapReduce) is a popular framework for data-intensive distributed computing of batch jobs. In particular, it is very useful for “cooking” massive raw data into useful information that is consumed by another storage system. In our system, Hadoop is used to parse continuously streaming tweets as well as weather data delivered in an XML format, based on a cluster that is built on 16 machines. The parsed data are then stored in HBase, which is commonly used as a “Hadoop storage”.

- *Front-End* implements a user interface of the recommendation system. Specifically, this component takes user inputs for querying, and delivers the inputs to the back-end. The query results returned from the back-end are then visualized through the front-end.

6.4.3 Data Processing

The travel recommendation system computes and maintains a set of 2D maps of Weekdays (D_i) x Weather-Labels (W_j). The cells in each map stores the mood probabilities computed by analyzing the data points of tweets mapped to the mood space, as described in Section 6.3. Figures 6.4(a),6.4(b) and 6.4(c) shows the visualization of these 2D maps, where each subfigure corresponds to a distinct weather label. The system manages seven different discs corresponding to seven days in a week (Mon, Tue, . . ., Sat, Sun). The outermost disc corresponds to Monday while innermost corresponds to Sunday. Each disc contains the different mood distributions computed through the data fusion process.

The system computes each entry of the discs using a massively parallel computing job. It employs map and reduce jobs in MapReduce (Hadoop) [DG04] to run the ETL-oriented processing in parallel. Algorithms 6.1 and 6.2 offer in detail the operations of mapper and reducer.

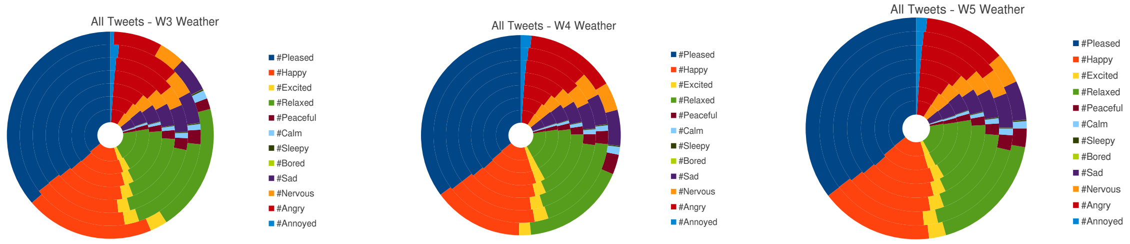
Algorithm 6.1: Mapper Job for Social and Sensor Data Fusion

```
procedure MAP($TweetId, Tweet$)
    ANEW[] {contains valence and arousal scores of set of words}
    $Tweet \rightarrow words[]$ {decompose Tweet into words}
    for $word_i \in words[]$ do
        if word in ANEW[] then
            $(val_i, ars_i) \leftarrow$ ANEW[$word_i$]
        else
            $(val_i, ars_i) \leftarrow (0,0)$
        end if
    end for
    $tweet\_val \leftarrow \frac{\sum_{val_i \neq 0} val}{num(val_i \neq 0)}$
    $tweet\_ars \leftarrow \frac{\sum_{ars_i \neq 0} ars}{num(ars_i \neq 0)}$
    mood: $M_i \leftarrow$ moodMap2DFn($tweet\_val, tweet\_ars$)
    location: $L \leftarrow$ locationOf($Tweet$)
    time: $t \leftarrow$ timeOf($Tweet$)
    Day: $D_k \leftarrow$ dayOf($Tweet$)
    Emit(($L$,$D_k$,$t$),($M_k$))
end procedure
```

The query processing in the recommendation system then uses the 2D mood map discs computed by the mapper and reducer. Specifically, when a user needs to know which would be the mood on a certain day and place, the system queries the WeatherUnderground API to obtain the weather forecast of the input day. At the same time, the system also queries the 2D structures to know the probabilities of mood states for that travel day. As shown, it is straightforward to add another dimension in the fusion.

Algorithm 6.2: Reducer Job for Social and Sensor Data Fusion

```
procedure REDUCER($Key, Value$)
    {Computes Mood Space Probability Distributions}
    WeatherMap[]{contains weather labels for different timestamps}
    (Location:$L$,Day:$D_k$,time:$t$) ← decompose($Key$)
    Weather Label: $W_j$ ← WeatherMap[$t$]
    Mood: $M_i$ ← Value
    increment(locationMoodMap[$W_j$][$D_k$][$M_i$], 1)
    return
end procedure
```



(a) Mood map for W3 weather (b) Mood map for W4 weather (c) Mood map for W5 weather

Figure 6.4: Probability distribution over mood-spaces are shown in (a), (b) and (c) corresponding to weathers W3, W4 and W5 respectively. In each mood map, there are seven discs and each disc corresponding to a weekday. Outer disc corresponds to Monday while the inner most disc corresponds to Sunday.

6.5 Experiments

6.5.1 Data Collection Process

In our current setting, we work with the Twitter social network API for obtaining the *social data*. We collect all the tweets corresponding to *London* location. In order to decide if a tweet is about a particular location, L_i, we use multiple features of the Twitter API. We consider a tweet is about a particular location if the tweet metadata has geo-tag information[1], or if the tweet user is from this place, or if the tweet text contains the location name. With these rules, we manage to obtain an approximate rate of 80-90 tweet messages per minute for the city of London, England.

We consider weather information at a particular location as *sensor data* in our data fusion setting. Specifically, we make use of services provided by WeatherUnderground[2], in order to periodically query the weather (W_j) of a particular location L_i. WeatherUnderground is a service that provides real-time weather information from nearly 32,000 weather stations around the world. The API provides wide variety of weather information like wind speed, wind direction, pressure, weather label etc. We are mainly concerned with the weather label. Some examples of the weather labels are *drizzle*, *rain*, *clear*, *thunderstorm* etc. We categorize the weather labels into 5 sets $\{W1, W2, W3, W4, W5\}$, as shown in Table 6.1, according to pleasantness. As we move from W1 to W5 the pleasantness of the weather

[1] Many smart phones provide this information automatically for the tweets posted using them.

[2] `http://www.wunderground.com`

W5	Clear	W3	Overcast Drizzle
W4	Scattered Clouds Partly Cloudy Mostly Cloudy	W2	Snow Fog
W3	Showers Rain Haze Rain	W1	Thunderstorm Thunderstorms and Rain Thunderstorms and Snow

Table 6.1: Categories of different Weather labels

increases. We collect weather information of *London*, once every 30 minutes under the assumption that weather stays same over this period.

We collect both the *social* and *sensor* data by deploying corresponding virtual sensors in the GSN. These social and weather virtual sensors contain all the rules,filtering conditions and rate controlling parameters for collecting the data needed for the fusion process. One can easily include other locations into our travel recommendation system just through adding corresponding virtual sensors to GSN.

The weather data and twitter social data collected using the GSN framework is stored HBase back-end deployed on a cluster of 16 machines. The extraction of metrics from social data, and the fusion process of social and sensor data is done through the use of various configurable Hadoop (Map-Reduce) jobs.

6.5.2 Dataset Statistics

We summarize the amounts of social and sensor data we collected over a period of 100 days for one particular location "*London*". The emotion expressed in a tweet might be related to different factors (weather, stock market influence, personal, work, product, event etc.). In order to focus on tweets related to weather we identify subset of the collected twitter dataset that is actually related to weather, and we refer to this subset as *weather-related* dataset. Given a tweet, we decide if a tweet is weather-related tweet using a set of weather related keywords. We summarize our results and observations corresponding to both the complete twitter dataset and weather-related twitter dataset. In Table 6.2 we summarize the statistics of the sizes of the datasets.

	Twitter Data	**Weather Data**
Complete Dataset		
Duration	28-April-2011 to 10-August-2011	
Number of Entries	12 Million	6600
Weather Related Entries	500000	6600
Training Dataset		
Duration	28-April-2011 to 20-June-2011	
Number of Entries	6.5 Million	3800
Weather Related Entries	300000	3800
Test Dataset		
Duration	21-June-2011 to 10-August-2011	
Number of Entries	5.5 Million	2800
Weather Related Entries	200000	2800

Table 6.2: Data Collection Characterization

Table 6.3 shows summarized view of the number of tweets we observed after binning them according to the weather labels shown in Table 6.1. In the table we observe 0 tweets for weather labels W1 & W2, as there were no thunderstorms or snow during the time window (April-August) in which we collected our tweets. Table 6.4 shows a uniform distribution of number of tweets collected on different weekdays.

	Weather Labels				
#Tweets	W1	W2	W3	W4	W5
All	0	0	1836460	5201270	5108473
Weather-Related	0	0	124048	211000	198645

Table 6.3: Tweets distribution w.r.t. Weather labels

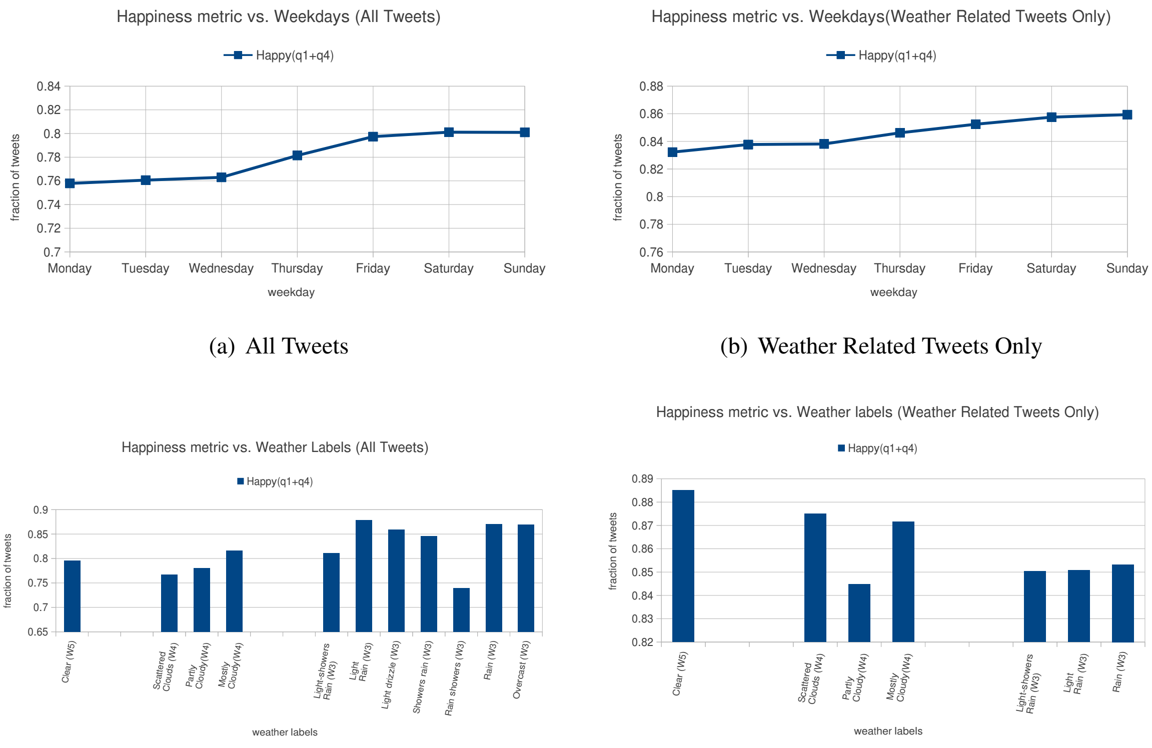
#Tweets	All	Weather-Related
Mon	2197788	83585
Tue	2532976	86485
Wed	1519632	70625
Thu	1585580	80324
Fri	1655613	74684
Sat	1413140	68048
Sun	1241474	69942

Table 6.4: Tweets distribution w.r.t. Weekdays

6.5.3 Observations

In this subsection we discuss the different observations we made regarding the mood metrics, and the correlation w.r.t. weekdays and weather labels. Some of the questions we asked were: *what is the happiness trend with respect to the weekdays? and what is the trend w.r.t to different weather conditions?* Even though our mood space is divided into 12 different mood spaces, for answering the above questions, we simplify our problem by considering all the moods in the quadrants Q1 and Q4 (valence $>$ 5) as *happy* and the moods in quadrants Q2 and Q3 (valence $\leq$ 5) will be termed as *sad*. Figures 6.5(a) and 6.5(b) show the fraction of *happy* tweets we observed on different weekdays $\{Mon, \dots, Sun\}$ irrespective of the weather conditions. Figure 6.5(a) represents the trend when the complete twitter data is taken into consideration, while Figure 6.5(b) corresponds only to weather related tweets. In either case, we observe that people in general are happier on the weekends $\{Fri, Sat, Sun\}$ compared to the weekdays $\{Mon, Tue, Wed, Thu\}$. Also we observe $Monday$ has the least fraction of *happy* tweets.

Next we tried to see any similar trends with respect to the different weather labels $\{W1, \dots, W5\}$ irrespective of the weekdays, whose results are shown in Figure 6.5(c) and Figure 6.5(d). For weather-related tweets, the results shown in Figure 6.5(d) we see that people are happiest on sunnier (W5) days, followed by cloudy (W4) days and least happy when it is raining (W3). On the contrary we did not see any clear trend when we consider the complete twitter data, as shown in Figure 6.5(c). It may be suggesting that for a place like London, the weather, per se, does not have significant impact on the mood of the public.



(a) All Tweets

(b) Weather Related Tweets Only

(c) All Tweets

(d) Weather Related Tweets Only

Figure 6.5: Fraction of tweets expressing happiness mood (all tweets in Q1 & Q4 quadrant). (a) shows happiness metric for the complete tweets dataset, while (b) corresponds to weather related tweets only. In both cases we observe the tweets on the weekends tend to appear more happier than the tweets on the weekdays. (c) shows happiness metric of all tweets w.r.t. different weather labels, while (d) concerns only weather related tweets. Only in the later case we see people are more happier on sunnier days.

6.5.4 Recommendation Validation

For travel recommendation system, when a user queries for a particular location in near future, adding to the future weather prediction of the location we would also try to predict the mood levels of the people. It is possible to make the prediction of near future events, based on the past history. One could imagine different prediction models. In our simplistic prediction model we summarize the statistics seen so far and we expect them to be valid for the future events. In order to evaluate the validity of this model, we divided the entire tweet dataset into two time windows, the training window and the test window.

We rely on two accuracy metrics to see if the statistics inferred on the training window (history) are still similar to the statistics observed in the test window (near future events). The first metric we rely on essentially compares two probability distributions, in our case the distributions over mood spaces described in Figure 6.1. We have two mood-spaces distributions corresponding to training and test windows. In order to see if they are from similar distribution we apply Chi-Square Goodness-of-Fit test [SC89]. We observed values of $\chi^2 = 0.0056$ (Weather Related Tweets) and $\chi^2 = 0.054$ (All Tweets). Such low values of χ^2 suggest that we accept the null hypothesis that both the probability distributions are very similar. Further the distributions are much more similar in the weather-related tweets compared to all-tweets case.

	Mon	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
W3	1.000	0.667	0.667	0.667	0.667	0.667	0.667
W4	0.667	0.667	0.667	0.667	0.667	0.667	0.667
W5	0.667	0.667	0.667	0.667	0.667	0.667	0.667

Table 6.5: Moods Overlap: Jaccard Similarity metric when considering the **complete dataset**

	Mon	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
W3	0.66667	1	0.66667	1	1	1	1
W4	0.66667	0.66667	0.66667	1	1	1	1
W5	1	1	1	1	1	1	1

Table 6.6: Moods Overlap: Jaccard Similarity metric when considering the **weather-related twitter dataset**

The second test we consider is the percentage of overlap between the top 5 moods predicted by the training window model and the ones we observe during the test window. We use Jaccard Similarity metric to quantify the overlap, which is defined among two sets as the ratio of their intersection size to the union size. The observed overlap metrics are shown in tables Table-6.5 and Table-6.6. In both the cases we observe a significant overlap between the predicted moods and observed moods. The mood labels we learn through our fusion process during the training window significantly overlap with the mood labels we observe during the test window.

Our fusion process of social and sensor data in the cloud, not only helped us understand the general trends of mood swings with respect to different weekdays and weather labels but also through simple models make accurate predictions. Relying on scalable cloud components, our fusion process can be readily expanded to many more locations with little effort. Through careful tuning of map-reduce jobs our fusion process can handle far more complex prediction models.

6.6 Conclusions

In this chapter, we presented profile construction by aggregating social and sensor metrics from the content corresponding to a location entity. As various smart applications rely on mobile cloud computing, fusing data in the cloud becomes an essential issue. Addressing this concern, we presented a data-fusion approach that blends representative data sources—social and sensor data corresponding to a location entity—commonly managed in mobile cloud applications. Specifically, we explored the potential of the data fusion by proposing a theoretical framework that enables to analyze tweet messages for extracting people's moods depending on day, weather, and location. We implemented the framework as a travel recommendation system that facilitates the fusion process over massively streaming data. The system is established upon several well-known cloud systems, allowing scalable data-fusion processing. We then discussed about various findings obtained from comprehensive experimental results using 12 million tweets as well as meteorological sensor readings, which demonstrate the effectiveness of our proposal.

Part V

Conclusions

Chapter 7

Conclusion and Future Work

> *Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning.*
>
> *Winston Churchill*

7.1 Conclusion

In this dissertation we have addressed several entity-related challenges. We presented a number of contributions that aid in linking the entity mentions across the web for creating a global knowledge graph of linked entities, thus facilitating wider adoption of the Semantic Web. We addressed following important entity-related problems: *Entity Resolution for Web Documents*, *Entity Matching in Twitter environments*, and *Entity Profiling*, which make a promising step towards realizing entity-oriented view of Semantic Web.

In the first part of the thesis, we proposed Entity Resolution methods for Web data collections, in particular to realize Web people search [YMA10b]. We studied the design of similarity assessment techniques. Our proposed method estimates the quality of available similarity values, for particular regions of the input and not globally, as the assessment techniques themselves produce results of different quality. Specifically it takes into account if some information is missing, which is very common in the context of Web documents. We demonstrated the effectiveness of these methods in our framework [YMA12b] using two real world datasets and showed promising results. Quality-aware similarity functions can be used in combination with other algorithmic frameworks as well. The systematic quality assessment and quality-aware combination technique results improved similarity values and also the overall performance of these algorithms.

Nowadays people are readily expressing themselves on microblogging platforms like Twitter. For many organizations, such messages are of great importance for many business decisions. In the second part of the thesis, we addressed the problem of entity-based matching of tweets through several techniques. We presented a simple Naive Bayes classifier, which relies on automatically or semi-automatically constructed profiles [YMA10a]. We then extended this basic technique in two ways. First, we developed a method that takes estimations of the general ambiguity level of the problem into account.

Second, we also introduced a technique [YMA11] that updates our company profiles actively from the twitter stream. We were able to analyze our techniques, find the sources of error in our profile construction techniques, and introduced methods to systematically address these problems [YMA12a]. Our experiments showed systematic improvements as we extend our classifier with the described techniques. Using our demo, TweetSpector [YMG$^+$12], we showed a prototype of our entity-based classification of tweets.

In the final part, we focused on the *Entity Profiling* problem, which is about summarizing the information related to an entity. We mainly focused our efforts on profiling an user entity [YGTA13, YCDA13] and a location entity [YSJA12, YJA12]. We presented a number of techniques for constructing user entity profiles [YCDA13], and evaluated their effectiveness for the tweet disambiguation task. Such user entity profiles present a summarized view of the user generated content across various social networks. We have also shown the importance of context in handling the tweet ambiguity. We used the user entity profiles to provide the missing context to the microposts, thus seeing an improved performance of the tweet classifier. Specifically, frequency-based features on Twitter and LLDA features on StackOverflow result-in user profiles that significantly improve effectiveness of disambiguation as compared to baseline approaches. *TripEneer* [YGTA13]: a travel plan recommendation application of user-entity social profile is presented. Finally we presented a data-fusion approach [YSJA12, YJA12] that blends representative data sources—social and sensor data related to a location entity—commonly managed in mobile cloud applications. Specifically, we explored the potential of the data fusion by proposing a theoretical framework that analyzes tweet messages for extracting people's moods depending on day, weather, and location. We implemented the framework as a travel recommendation system that facilitates the fusion process over massively streaming data.

7.2 Future Directions

There are still a number of open challenges that need to be addressed to fully realize entity-oriented view of the Semantic Web. Significant work needs to be done for development of entity extraction tools that can work reliably on microblogging kind of media, as this media is becoming as relevant as other news and social media. Keywords based indexes and inverted document indexes have helped keyword based search systems. Newer innovations are needed in designing indexes and back-end architectures for supporting entity-based and semantic search systems. There is also need for research efforts in designing user interfaces for such systems.

We recognize that the work described in this thesis can be strengthened in a number of ways and specifically, we suggest the following as future work.

7.2.1 Entity Resolution for Web Documents

In relation to Entity Resolution techniques for Web Documents (Chapter 3), we propose following improvements as future work.

- For efficient ER techniques our framework defined regions (Section 3.2.3) in the range of similarity functions and relied on supervised ML techniques for estimating the accuracies of similarity functions in these regions. In our future work we would like to find dataset independent ways of defining regions for accuracy estimations. We also plan to address the effect of incomplete information available in the Web pages on the accuracy of the similarity functions even more directly, by considering entropy based metrics, similar to [CMBHA08].

- The efficiency of the combination technique (Algorithm 3.1, Step 4) depends on the quality and diversity of the similarity functions participating. We would like to explore models to compute the theoretical best achievable performance. As our techniques are based on supervised machine learning (ML) approaches, there is an involved cost of obtaining the training examples. We would like to explore semi-supervised and unsupervised ML approaches to this problem. We could possibly exploit the relative importance of the entities.

- Once we cluster the web documents based on the real world entity, we plan to explore techniques to summarize the entity representing the cluster. We need to study the impact on summarization when we add or remove documents from the cluster.

7.2.2 Entities in Twitter Streams

With respect to Entities in Twitter Streams (Chapter 4), we can extend our work in the following directions:

- The core of our entity-based classifiers of tweets are the entity profiles. We relied on a number of information sources – mainly static – (homepages, Google sets, Wikipedia disambiguation pages, etc) for constructing the initial company entity profile (Section 4.2.2.2). As new products and media information about companies are released, it is important that company-entity profile is up-to-date so that it does not miss relevant tweets. We need cleaner ways of integrating any such dynamic information into the company profiles. One possible way is by actively following the news section of the company.

- Human feedback on tweets involves cost. In future work we plan to optimize on the accuracy of the classifier by keeping the human feedback costs low by selecting a subset of those tweets, on which human feedback would benefit the classifier the most. Active learning techniques from machine learning could be exploited for this purpose.

- *TweetSpector* (Section 4.8) is shown as prototype for company entity based classification of tweets. In future, it can be extended to other types of interesting entities (TV-shows, sport teams, etc.) and sub-entities (sub-group in a company, or a specific product in a company, etc.) for which similar entity profiles could be constructed. We also plan to provide *TweetSpector* as a webservice API, using which interested company entities can get relevant real-time tweets on to their websites.

7.2.3 Entity Profiling and Applications

In reference to entity profiling and applications (Chapter 5 & Chapter 6), we propose following directions for the future work.

- In our efforts to construct an entity profile of an user we treated all the content (from Twitter & StackOverflow) similarly. However, as a user has multiple facets, all the content on bigger social networks (like Facebook) is not equal. The content corresponds to his different interests:

sports, travel, movies, books, etc. One could explore complex model for user entity profile, which has all this facets. Each facet is built on the corresponding content. Interesting recommendation applications can request the user for a particular relevant facet.

- In future we would like to enrich the user entity profiles by making use of additional information sources : lists, the social network connections, and the activity flow (URLs shared and propagated). These additional sources could be used to identify the experts w.r.t. topics. One could also recommend new users and content relevant to a particular user based on his entity profile.
- In *TripEneer* (Section 5.6) application, we presented hybrid ranking as the one which combines various other rankings. In the future work, we plan to infer the weights for various components based on the users activity, rather than manually setting them. We could also infer them based on an interactive questionnaire.
- Using cloud technologies we computed aggregated values by mining tweets related to a particular location (Chapter 6). We would like to explore models where batch updates could be done to the aggregated values corresponding to social metrics of a location entity profile. With the scalable fusion framework, we would like to extend our study, the impact of weather has on peoples' moods [Pul], for many more cities.

Bibliography

[AAG$^+$10] E. Amigo, J. Artiles, J. Gonzalo, D. Spina, B. Liu, and A. Corujo. WePS3 Evaluation Campaign: Overview of the On-line Reputation Management Task. In *2nd Web People Search Evaluation Workshop (WePS 2010), CLEF 2010 Conference, Padova Italy*, 2010. 26, 65, 71, 76, 77, 82, 84, 92, 104, 107

[ACG02] Rohit Ananthakrishna, Surajit Chaudhuri, and Venkatesh Ganti. Eliminating fuzzy duplicates in data warehouses. In *Proceedings of the 28th international conference on Very Large Data Bases*, VLDB '02, pages 586–597. VLDB Endowment, 2002. 19

[ACN08] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *J. ACM*, 55(5):23:1–23:27, November 2008. 17, 19

[AGAV11] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. Combining evaluation metrics via the unanimous improvement ratio and its application to clustering tasks. *J. Artif. Int. Res.*, 42(1):689–718, September 2011. 75

[AGHT11] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. Analyzing user modeling on twitter for personalized news recommendations. In *Proceedings of the 19th international conference on User modeling, adaption, and personalization*, UMAP'11, pages 1–12, Berlin, Heidelberg, 2011. Springer-Verlag. 23, 27, 29, 105, 106

[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*, SIGMOD '10, pages 783–794, New York, NY, USA, 2010. ACM. 17

[AHK11] Fabian Abel, Eelco Herder, and Daniel Krause. Extraction of professional interests from social web profiles. In *Proc. AUM 2011 - Workshop on Augmenting User Models with Real World Experiences to Enhance Personalization and Adaptation, co-located with UMAP 2011.*, 2011. 27, 29, 105, 106

[AHS06] Karl Aberer, Manfred Hauswirth, and Ali Salehi. The Global Sensor Networks middleware for efficient and flexible deployment and interconnection of sensor networks. Technical report, 2006. Sumitted to ACM/IFIP/USENIX 7th International Middleware Conference. 114, 120

[AM12] B. Adams and G. McKenzie. Frankenplace: An application for similarity-based place search. *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media*, 2012. 108

[ARS09] Arvind Arasu, Christopher Ré, and Dan Suciu. Large-scale deduplication with constraints using dedupalog. In *Proceedings of the 2009 IEEE International Conference on Data Engineering*, ICDE '09, pages 952–963, Washington, DC, USA, 2009. IEEE Computer Society. 17, 18

[AXV$^+$11] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 30–38, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. 25, 26

[BAdR06] Krisztian Balog, Leif Azzopardi, and Maarten de Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 43–50, New York, NY, USA, 2006. ACM. 27

[BAdR08] Krisztian Balog, Leif Azzopardi, and Maarten de Rijke. Resolving person names in web people search. In Irwin King and Ricardo A. Baeza-Yates, editors, *Weaving Services and People on the World Wide Web*, pages 301–323. Springer, 2008. 55

[BBC04] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation Clustering. *Machine Learning*, 56(1-3):89–113, 2004. 17, 41, 56

[BBS05] Mikhail Bilenko, Sugato Basu, and Mehran Sahami. Adaptive product normalization: Using online learning for record linkage in comparison shopping. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, ICDM '05, pages 58–65, Washington, DC, USA, 2005. IEEE Computer Society. 17

[BCFM98] Andrei Z. Broder, Moses Charikar, Alan M. Frieze, and Michael Mitzenmacher. Min-wise independent permutations (extended abstract). In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, STOC '98, pages 327–336, New York, NY, USA, 1998. ACM. 20

[BG05] José Barateiro and Helena Galhardas. A survey of data quality tools. *Datenbank-Spektrum*, 14:15–21, 2005. 21

[BG06] Indrajit Bhattacharya and Lise Getoor. A latent dirichlet model for unsupervised entity resolution. In *SIAM INTERNATIONAL CONFERENCE ON DATA MINING*, 2006. 17, 18

[BG07] Indrajit Bhattacharya and Lise Getoor. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1), 2007. 17

[BGB08] Yaxin Bi, Jiwen Guan, and David Bell. The combination of multiple classifiers using an evidental reasoning approach. *Artificial Inelligence*, 172:1731–1751, 2008. 23, 56

[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. 15, 16, 17, 21, 33, 54

[BGX$^+$10] Aaron Beach, Mike Gartrell, Xinyu Xing, Richard Han, Qin Lv, Shivakant Mishra, and Karim Seada. Fusing mobile, sensor, and social data to fully enable context-aware computing. In *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications*, HotMobile '10, pages 60–65, New York, NY, USA, 2010. ACM. 116

[BHLZ10] Alina Beygelzimer, Daniel Hsu, John Langford, and Tong Zhang. Agnostic active learning without constraints. *CoRR*, abs/1006.2588, 2010. 17

[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*, KDD '12, pages 1131–1139, New York, NY, USA, 2012. ACM. 17

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. 83

[BKM06] Mikhail Bilenko, Beena Kamath, and Raymond J. Mooney. Adaptive blocking: Learning to scale up record linkage. In *Proceedings of the Sixth International Conference on Data Mining*, ICDM '06, pages 87–96, Washington, DC, USA, 2006. IEEE Computer Society. 20

[BL99] Margaret M. Bradley and Peter J. Lang. Affective norms for english words (anew): Instruction manual and affective ratings. 1999. 116, 118

[BM03] Mikhail Bilenko and Raymond J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 39–48, 2003. 16, 23, 56

[BM05] Ron Bekkerman and Andrew McCallum. Disambiguating Web appearances of people in a social network. In *Proceedings of the 14th international conference on World Wide Web*, pages 463–470, 2005. 26, 48, 51, 83

[BMG10] Matthias Broecheler, Lilyana Mihalkova, and Lise Getoor. Probabilistic similarity logic. In *Conference on Uncertainty in Artificial Intelligence*, 2010. 18

[BMZ10] Johan Bollen, Huina Mao, and Xiao-Jun Zeng. Twitter mood predicts the stock market. *ArXiv e-prints*, October 2010. 25, 115

[BNJ03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, January 2003. 27, 28, 106

[BPSV09] Paolo Bouquet, Themis Palpanas, Heiko Stoermer, and Massimiliano Vignolo. A conceptual model for a web-scale entity name system. In *The Semantic Web, Fourth Asian Conference, ASWC 2009*, number 5926 in LNCS, pages 46–60. Springer, 2009. 22, 55

[BSG07] Paolo Bouquet, Heiko Stoermer, and Daniel Giacomuzzi. Okkam: Enabling a web of entities. In *I3*, 2007. 5, 7

[BT06] David G. Brizan and Abdullah U. Tansel. A Survey of Entity Resolution and Record Linkage Methodologies. *Communications of the IIMA*, 6(3), 2006. 92

[CGM05] Surajit Chaudhuri, Venkatesh Ganti, and Rajeev Motwani. Robust Identification of Fuzzy Duplicates. In *Proceedings of the 21st International Conference on Data Engineering (ICDE)*, pages 865–876, 2005. 14, 17, 19, 21, 33, 34, 54

[Chr08] Peter Christen. Automatic record linkage using seeded nearest neighbour and support vector machine classification. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 151–159, New York, NY, USA, 2008. ACM. 16

[Cit] Citeseer. http://citeseerx.ist.psu.edu/. 7

[CKGS06] Chia-Hui Chang, Mohammed Kayed, Moheb Ramzy Girgis, and Khaled Shaalan. A survey of web information extraction systems. 2006. 92

[CKLS01] Munir Cochinwala, Verghese Kurien, Gail Lalk, and Dennis Shasha. Efficient data reconciliation. *Inf. Sci.*, 137(1-4):1–15, 2001. 16

[CKM07] Zhaoqi Chen, Dmitri V. Kalashnikov, and Sharad Mehrotra. Adaptive graphical approach to entity resolution. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 204–213, 2007. 33

[CKM09] Zhaoqi Chen, Dmitri V. Kalashnikov, and Sharad Mehrotra. Exploiting context analysis for combining multiple entity resolution systems. In *Proceedings of the 35th SIGMOD international conference on Management of data*, pages 207–218, 2009. 16, 23, 26, 51, 56, 83

[CLY07] Sungha Choi, Byungwoo Lee, and Jihoon Yang. Ensembles of region based classifiers. In *CIT '07: Proceedings of the 7th IEEE International Conference on Computer and Information Technology*, pages 41–46, Washington, DC, USA, 2007. IEEE Computer Society. 83

[CM07] Philippe Cudré-Mauroux. idmesh: Decentralized identity management for the declarative web (extended version). 2007. 21

[CMBHA08] Philippe Cudré-Mauroux, Adriana Budura, Manfred Hauswirth, and Karl Aberer. PicShark: mitigating metadata scarcity through large-scale P2P collaboration. *The VLDB Journal : The International Journal on Very Large Data Bases*, 17(6):1371–1384, November 2008. 57, 131

[CMHJ$^+$09] Philippe Cudré-Mauroux, Parisa Haghani, Michael Jost, Karl Aberer, and Hermann de Meer. idMesh: Graph-Based Disambiguation of Linked Data. In *Proceedings of the 18th International World Wide Web Conference (WWW'09)*, 2009. 22, 54

[Coh01] William Cohen. Learning to match and cluster entity names. In *In ACM SIGIR-2001 Workshop on Mathematical/Formal Methods in Information Retrieval*, 2001. 15

[Cor] Cora. Cora citations dataset. http://people.cs.umass.edu/ mccallum/data.html. 7

[CRF03] W. Cohen, P. Ravikumar, and S. Fienberg. A comparison of string metrics for matching names and records. In *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web*, pages 73–78, August 2003. 15

[CSGK07] Surajit Chaudhuri, Anish Das Sarma, Venkatesh Ganti, and Raghav Kaushik. Leveraging aggregate constraints for deduplication. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 437–448, 2007. 19, 21, 54

[CST00] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000. 47

[Cun02] H. Cunningham. GATE, a General Architecture for Text Engineering. *Computers and the Humanities*, 36:223–254, 2002. 38

[CVSPO10] Miguel Angel García Cumbreras, Manuel García Vega, Fernando Martínez Santiago, and José M. Perea-Ortega. SINAI at WePS-3: Online Reputation Management. In Martin Braschler, Donna Harman, and Emanuele Pianta, editors, *CLEF (Notebook Papers/LABs/Workshops)*, 2010. 26, 74, 84

[CWH$^+$07] Aron Culotta, Michael Wick, Robert Hall, Matthew Marzilli, and Andrew McCallum. Canonicalization of database records using adaptive similarity measures. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 201–209, New York, NY, USA, 2007. ACM. 17

[CWS12] Lu Chen, Wenbo Wang, and AmitP. Sheth. Are twitter users equal in predicting elections? a study of user groups in predicting 2012 u.s. republican presidential primaries. In Karl Aberer, Andreas Flache, Wander Jager, Ling Liu, Jie Tang, and Christophe Guéret, editors, *Social Informatics*, volume 7710 of *Lecture Notes in Computer Science*, pages 379–392. Springer Berlin Heidelberg, 2012. 26

[Dap] Dapper. http://www.dapper.net/. 4

[DBES09] Xin Luna Dong, Laure Berti-Equille, and Divesh Srivastava. Integrating conflicting data: the role of source dependence. *Proc. VLDB Endow.*, 2(1):550–561, August 2009. 17

[DBp] DBpedia. http://www.http://dbpedia.org/. 4

[DFD11] Ovidiu Dan, Junlan Feng, and Brian Davison. A Bootstrapping Approach to Identifying Relevant Tweets for Social TV. 2011. 24, 84

[DG04] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. Berkeley, CA, USA, 2004. USENIX Association. 20, 121

[DHM05] Xin Dong, Alon Halevy, and Jayant Madhavan. Reference reconciliation in complex information spaces. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 85–96, New York, NY, USA, 2005. ACM Press. 15, 17, 22, 33, 54

[DPH$^+$09] Gérard Dray, Michel Plantié, Ali Harb, Pascal Poncelet, Mathieu Roche, and François Trousset. Opinion Mining From Blogs. *IJCISIM'09: International Journal of Computer Information Systems and Industrial Management Applications*, 1:205–213, 2009. 24, 25

[DSJMB12] Anish Das Sarma, Ankur Jain, Ashwin Machanavajjhala, and Philip Bohannon. An automatic blocking mechanism for large-scale de-duplication tasks. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, pages 1055–1064, New York, NY, USA, 2012. ACM. 20

[EC08] Micha Elsner and Eugene Charniak. You talking to me? a corpus and algorithm for conversation disentanglement. In Kathleen McKeown, Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui, editors, *ACL*, pages 834–842. The Association for Computer Linguistics, 2008. 17

[ES07] J. Euzenat and P. Shvaiko. *Ontology matching*. Springer, 2007. 22, 55

[ES09] Micha Elsner and Warren Schudy. Bounding and comparing methods for correlation clustering beyond ilp. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, ILP '09, pages 19–27, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. 17

[Fan08] Wenfei Fan. Dependencies revisited for improving data quality. In *Proceedings of the twenty-seventh ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '08, pages 159–170, New York, NY, USA, 2008. ACM. 18

[Fre] FreeBase. http://www.freebase.com/. 4

[FS69] Ivan P. Fellegi and Alan B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969. 15, 16, 33, 54

[GAC$^+$10] Wojciech Galuba, Karl Aberer, Dipanjan Chakraborty, Zoran Despotovic, and Wolfgang Kellerer. Outtweeting the Twitterers - Predicting Information Cascades in Microblogs. In *3rd Workshop on Online Social Networks (WOSN'10)*, 2010. 23, 41

[GAHY12] Qi Gao, Fabian Abel, Geert-Jan Houben, and Yong Yu. A comparative study of users' microblogging behavior on sina weibo and twitter. In *Proceedings of the 20th international conference on User Modeling, Adaptation, and Personalization*, UMAP'12, pages 88–101, Berlin, Heidelberg, 2012. Springer-Verlag. 23, 27, 28, 29, 105, 106

[GAT] GATE. Gate information extraction. http://gate.ac.uk/ie/. 5

[Gov] Government. Semantic web data. http://semanticweb.com/category/government. 4

[GS96] Ralph Grishman and Beth Sundheim. Message understanding conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics - Volume 1*, COLING '96, pages 466–471, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics. 5

[GS09] Rahul Gupta and Sunita Sarawagi. Answering table augmentation queries from unstructured lists on the web. *Proc. VLDB Endow.*, 2(1):289–300, August 2009. 16

[GSS07] Namrata Godbole, Manjunath Srinivasaiah, and Steven Skiena. Large-scale sentiment analysis for news and blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, 2007. 24, 25

[HAG$^+$12] Liangjie Hong, Amr Ahmed, Siva Gurumurthy, Alexander J. Smola, and Kostas Tsioutsiouliklis. Discovering geographical topics in the twitter stream. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 769–778, New York, NY, USA, 2012. ACM. 24

[HBa] Hbase. http://hbase.apache.org. 114, 120

[HBS10] John Hannon, Mike Bennett, and Barry Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 199–206, New York, NY, USA, 2010. ACM. 27, 29, 105, 106

[HD10] Liangjie Hong and Brian D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pages 80–88, New York, NY, USA, 2010. ACM. 27, 28, 105

[Hec96] David Heckerman. A tutorial on learning with bayesian networks. Technical report, Learning in Graphical Models, 1996. 64, 97

[HFC$^+$08] Jian Hu, Lujun Fang, Yang Cao, Hua-Jun Zeng, Hua Li, Qiang Yang, and Zheng Chen. Enhancing text clustering by leveraging Wikipedia semantics. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 179–186, 2008. 38, 48, 49

[HL04] Minqing Hu and Bing Liu. Mining opinion features in customer reviews. In *Proceedings of the 19th national conference on Artifical intelligence*, AAAI'04, pages 755–760. AAAI Press, 2004. 24

[HMOS12] John Hannon, Kevin McCarthy, Michael P. O'Mahony, and Barry Smyth. A multi-faceted user model for twitter. In Judith Masthoff, Bamshad Mobasher, Michel C. Desmarais, and Roger Nkambou, editors, *UMAP*, volume 7379 of *Lecture Notes in Computer Science*, pages 303–309. Springer, 2012. 23, 27, 29, 105, 106

[HS95] Mauricio A. Hernández and Salvatore J. Stolfo. The merge/purge problem for large databases. In *SIGMOD '95: Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, pages 127–138, New York, NY, USA, 1995. ACM. 15, 20, 33, 54

[HS09] Laura M. Haas and Aya Soffer. New challenges in information integration. *Data Warehousing and Knowledge Discovery*, 2009. 13

[HSW07] Thomas N. Herzog, Fritz J. Scheuren, and William E. Winkler. *Data Quality and Record Linkage Techniques*. Springer Publishing Company, Incorporated, 1st edition, 2007. 16

[INE] INEX. Initiative for the evaluation of xml retrieval: Entity track. http://www.springerreference.com/docs/html/chapterdbid/63287.html. 8

[Inf] Zoom Info. http://www.zoominfo.com/. 4

[IVE07] Panagiotis G. Ipeirotis, Vassilios S. Verykios, and Ahmed K. Elmagarmid. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 19(1):1–16, January 2007. 33, 54

[Jar89] Matthew A. Jaro. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association*, 84(406):414–420, 1989. 14

[Jar95] M. A. Jaro. Probabilistic linkage of large public health data file. In *Statistics in Medicine*, volume 14, pages 491–498, 1995. 14

[JRT10] Nikhil Johri, Dan Roth, and Yuancheng Tu. Experts' retrieval with multiword-enhanced author topic model. In *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search*, SS '10, pages 10–18, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. 27

[JZSC09] B.J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology*, 2009. 23, 26, 56, 83

[Kal10] Paul Kalmar. Bootstrapping Websites for Classification of Organization Names on Twitter. In Martin Braschler, Donna Harman, and Emanuele Pianta, editors, *CLEF (Notebook Papers/LABs/Workshops)*, 2010. 26, 74, 84

[KCMN08] Dmitri V. Kalashnikov, Zhaoqi Chen, Sharad Mehrotra, and Rabia Nuray. Web people search via connection analysis. *IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE)*, 20(11), November 2008. 22, 26, 51, 54, 83

[Kle99] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, sep 1999. 3

[KM06] Dmitri V. Kalashnikov and Sharad Mehrotra. Domain-independent data cleaning via analysis of entity-relationship graph. *ACM Transactions on Database Systems*, 31(2), 2006. 17, 21, 22, 54

[KML13] Jeon-Hyung Kang, Jun Ma, and Yan Liu. Transfer topic modeling with ease and scalability. *CoRR*, abs/1301.5686, 2013. 29, 107

[KR10] Hanna Köpcke and Erhard Rahm. Frameworks for entity matching: A comparison. *Data and Knowledge Engineering*, 69(2):197–210, February 2010. 54

[KRYL02] Sang-Bum Kim, Hae-Chang Rim, DongSuk Yook, and Heui-Seok Lim. Effective methods for improving naive bayes text classifiers. In Mitsuru Ishizuka and Abdul Sattar, editors, *PRICAI 2002: Trends in Artificial Intelligence*, volume 2417 of *Lecture Notes in Computer Science*, pages 479–484. Springer Berlin / Heidelberg, 2002. 67

[KSS06] Nick Koudas, Sunita Sarawagi, and Divesh Srivastava. Record linkage: similarity measures and algorithms. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, SIGMOD '06, pages 802–803, New York, NY, USA, 2006. ACM. 15

[KTF09] U. Kang, Charalampos E. Tsourakakis, and Christos Faloutsos. Pegasus: A peta-scale graph mining system implementation and observations. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, ICDM '09, pages 229–238, Washington, DC, USA, 2009. IEEE Computer Society. 21

[Lew98] David D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. pages 4–15. Springer Verlag, 1998. 64, 97

[lin] http://www.123people.com/. 5

[LMB$^+$13] Sian Lindley, Cathy Marshall, Richard Banks, Abigail Sellen, and Tim Regan. Rethinking the web as a personal archive. Proceedings of the 2013 international conference on World Wide Web (WWW 2013), May 2013. 29, 106

[LOIP10] Tom Lovett, Eamonn O'Neill, James Irwin, and David Pollington. The calendar as a sensor: analysis and improvement using data fusion with social networks and location. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, Ubicomp '10, pages 3–12, New York, NY, USA, 2010. ACM. 116

[LWH$^+$12] Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun, and Bu-Sung Lee. Twiner: named entity recognition in targeted twitter stream. In *SIGIR'12*, 2012. 26, 28, 105

[LY01] Rujie Liu and Baozong Yuan. Multiple classifiers combination by clustering and selection. *Information Fusion*, 2(3):163–168, 2001. 23, 55

[LZ12] Bing Liu and Lei Zhang. A survey of opinion mining and sentiment analysis. *Mining Text Data*, 2012. 92

[Mah] Mahout. Mahout: Scalable machine learning and data mining. http://mahout.apache.org/. 20

[MBB$^+$10] Zoltán Miklós, Nicolas Bonvin, Paolo Bouquet, Michele Catasta, Daniele Cordioli, Peter Fankhauser, Julien Gaugaz, Ekaterini Ioannou, Hristo Koshutanski, Antonio Mana, Claudia Niederée, Themis Palpanas, and Heiko Stoermer. From Web Data to Entities and Back. In *The 22nd International Conference on Advanced Information Systems Engineering (CAiSE'10)*, volume 6051 of *LNCS*, pages 302–316. Springer, 2010. 34, 83

[MBB$^+$11] Adam Marcus, Michael S. Bernstein, Osama Badar, David R. Karger, Samuel Madden, and Robert C. Miller. Twitinfo: Aggregating and visualizing microblogs for event exploration. 2011. 25, 115

[MBGM06] David Menestrina, Omar Benjelloun, and Hector Garcia-Molina. Generic Entity Resolution with Data Confidences. In *In First International VLDB Workshop on Clean Databases*, 2006. 21, 33, 54

[MDM07] Donald Metzler, Susan Dumais, and Christopher Meek. Similarity Measures for Short Segments of Text. In *Advances in Information Retrieval*, volume 4425 of *LNCS*, pages 16–27, 2007. 47

[ME96] Alvaro E. Monge and Charles Elkan. The field matching problem: Algorithms and applications. In *KDD*, pages 267–270, 1996. 14

[ME97] Alvaro E. Monge and Charles Elkan. An efficient domain-independent algorithm for detecting approximately duplicate database records. In *DMKD*, pages 0–, 1997. 14

[MF] Micro-Formats. http://microformats.org. 4

[Mj03] H. Müller and j. Problems, methods and challenges in comprehensive data cleansing. Technical Report HUB-IB-164, Humboldt-Universität zu Berlin, Institut für Informatik, 2003. 13

[MK06] Matthew Michelson and Craig A. Knoblock. Learning blocking schemes for record linkage. In *Proceedings of the 21st national conference on Artificial intelligence - Volume 1*, AAAI'06, pages 440–445. AAAI Press, 2006. 20

[MNU00] Andrew McCallum, Kamal Nigam, and Lyle H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 169–178, New York, NY, USA, 2000. ACM. 15, 20

[MRS08] Christopher D. Manning, Parbhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. 38, 48

[MW03] A. Mccallum and B. Wellner. Toward conditional models of identity uncertainty with application to proper noun coreference. 2003. 15, 17, 18

[MWK$^+$11] Adam Marcus, Eugene Wu, David Karger, Samuel Madden, and Robert Miller. Human-powered sorts and joins. *Proc. VLDB Endow.*, 5(1):13–24, September 2011. 17

[MWL$^+$12] Xinfan Meng, Furu Wei, Xiaohua Liu, Ming Zhou, Sujian Li, and Houfeng Wang. Entity-centric topic-oriented opinion summarization in twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 379–387, New York, NY, USA, 2012. ACM. 26

[NC02] Vincent Ng and Claire Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 104–111, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. 17

[Nep] Nepomuk. Nepomuk: Social semantic desktop. http://nepomuk.semanticdesktop.org. 4

[NER] NER. Named entity recognition (ner). http://en.wikipedia.org/wiki/Named-entity_recognition. 6

[NKAJ59] H. B. Newcombe, J. M. Kennedy, S. J. Axford, and A. P. James. Automatic linkage of vital records. *Science*, 130:954–959, October 1959. 15

[NS07] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1), January 2007. 92

[NSV11] Meena Nagarajan, Amit Sheth, and Selvam Velmurugan. Citizen sensor data mining, social media analytics and development centric web applications. In *WWW '11*, pages 289–290, New York, NY, USA, 2011. ACM. 115, 116

[Ope] OpenCalais. http://www.opencalais.com/. 4, 38

[OWL] OWL. Web ontology language. http://www.w3.org/2001/sw/wiki/OWL. 4

[OZHM13] Niels Ott, Ramon Ziai, Michael Hahn, and Detmar Meurers. Comet: Integrating different levels of linguistic modeling for meaning assessment. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval)*, GA, USA, 2013. 92

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. 3

[PC11] Aditya Pal and Scott Counts. Identifying topical authorities in microblogs. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 45–54, New York, NY, USA, 2011. ACM. 27, 28, 105

[PLV02] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. 24

[PMM$^+$03] Hanna Pasula, Bhaskara Marthi, Brian Milch, Stuart Russell, and Ilya Shpitser. Identity uncertainty and citation matching. In *In NIPS*. MIT Press, 2003. 18

[PP10] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA). 23, 25, 26, 56, 83, 115

[PRMB12] Aditya Pal, Vibhor Rastogi, Ashwin Machanavajjhala, and Philip Bohannon. Information integration over time in unreliable and uncertain environments. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 789–798, New York, NY, USA, 2012. ACM. 17

[PTPCR11] Fernando Perez-Tellez, David Pinto, John Cardiff, and Paolo Rosso. On the Difficulty of Clustering Microblog Texts for Online Reputation Management. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, pages 146–152, Portland, Oregon, June 2011. Association for Computational Linguistics. 24, 83

[Pul] Twitter Pulse. Pulse of the nation. http://www.ccs.neu.edu/home/amislove/twittermood/. 25, 115, 132

[RC04] Pradeep Ravikumar and William W. Cohen. A hierarchical graphical model for record linkage. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, UAI '04, pages 454–461, Arlington, Virginia, United States, 2004. AUAI Press. 16

[RCME11] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. 28, 105

[RDF] RDF. Resource description framework. http://www.w3.org/RDF/. 4, 8

[RDG11] Vibhor Rastogi, Nilesh Dalvi, and Minos Garofalakis. Large-scale collective entity matching. *Proc. VLDB Endow.*, 4(4):208–218, January 2011. 21

[RDL10] Daniel Ramage, Susan Dumais, and Dan Liebling. Characterizing microblogs with topic models. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*. AAAI, 2010. 28, 105, 106

[RHNM09] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 248–256, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. 27, 28, 92, 96, 106

[RMCS12] Vibhor Rastogi, Ashwin Machanavajjhala, Laukik Chitnis, and Anish Das Sarma. Finding connected components on map-reduce in logarithmic rounds. *CoRR*, abs/1203.5387, 2012. 21

[RMZ+11] A. Rosi, M. Mamei, F. Zambonelli, S. Dobson, G. Stevenson, and Juan Ye. Social sensors and pervasive services: Approaches and perspectives. In *Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 525 –530, march 2011. 115, 116

[SB02] Sunita Sarawagi and Anuradha Bhamidipaty. Interactive deduplication using active learning. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–278, New York, NY, USA, 2002. ACM. 15, 17

[SC89] G W Snedecor and W G Cochran. *Statistical Methods*. Iowa State University Press, 1989. 125

[SCS09] Harish Srinivasan, John Chen, and Rohini Srihari. Cross document person name disambiguation using entity profiles. Association for the Advancement of Artificial Intelligence, 2009. 29, 106

[SD06] Parag Singla and Pedro Domingos. Entity resolution with markov logic. In *In ICDM*, pages 572–582. IEEE Computer Society Press, 2006. 18

[SE10] Giovanni Seni and John Elder. *Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions*. Morgan and Claypool Publishers, 2010. 22, 55

[Sea] Zaba Search. http://www.zabasearch.com/. 5

[Sem] Semantify. Semantify. http://dapper.net/semantify/. 4

[SFD+10] Bharath Sriram, David Fuhry, Enngin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. Short text classification in twitter to improve information filtering. In *Proceedings of the ACM SIGIR 2010 Posters and Demos*. ACM, 2010. 23, 24, 56, 83

[SGC02] Alexander Strehl, Joydeep Ghosh, and Claire Cardie. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002. 23, 55

[SLD05] Warren Shen, Xin Li, and AnHai Doan. Constraint-based entity matching. In *AAAI*, pages 862–867, 2005. 17, 18, 19

[SNL01] Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. A machine learning approach to coreference resolution of noun phrases. *Comput. Linguist.*, 27(4):521–544, December 2001. 15, 17

[SOM10] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. WWW '10, pages 851–860, New York, NY, USA, April 2010. ACM. 115

[SPA] SPARQL. Sparql query language for rdf. http://www.w3.org/TR/rdf-sparql-query/. 4

[SST+09] Jagan Sankaranarayanan, Hanan Samet, Benjamin E. Teitler, Michael D. Lieberman, and Jon Sperling. Twitterstand: news in tweets. In *GIS '09: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 42–51, New York, NY, USA, 2009. ACM. 23, 24, 56, 83

[TB10] Manos Tsagkias and Krisztian Balog. The university of amsterdam at weps3. In Martin Braschler, Donna Harman, and Emanuele Pianta, editors, *CLEF (Notebook Papers/LABs/Workshops)*, 2010. 26, 27, 74, 84

[TBP11] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment in twitter events. *J. Am. Soc. Inf. Sci. Technol.*, 62:406–418, February 2011. 25, 115

[Tex] Textwise. http://www.textwise.com/. 4

[TKM02] Sheila Tejada, Craig A. Knoblock, and Steven Minton. Learning domain-independent string transformation weights for high accuracy object identification. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 350–359, New York, NY, USA, 2002. ACM. 15, 17

[TKW10] Bilyana Taneva, Mouna Kacimi, and Gerhard Weikum. Gathering and ranking photos of named entities with high precision, high recall, and diversity. In Brian D. Davison, Torsten Suel, Nick Craswell, and Bing Liu, editors, *WSDM*, pages 431–440. ACM, 2010. 24, 83

[TMT] Stanford TMT. Stanford topic modeling toolbox. http://nlp.stanford.edu/software/tmt/tmt-0.4/. 95, 96

[TRE] TREC. Text retrieval confernces. http://trec.nist.gov/. 5, 8

[TSSW10] A. Tumasjan, T.O. Sprenger, P.G. Sandner, and I.M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pages 178–185, 2010. 26

[Tur02] Peter D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. 24

[TY94] Liang Thow-Yick. The basic entity model: a fundamental theoretical model of information and information processing. *Inf. Process. Manage.*, 30(5):647–661, 1994. 15

[UIM] UIMA. Apache uima. http://uima.apache.org/. 5

[VCL10] Rares Vernica, Michael J. Carey, and Chen Li. Efficient parallel set-similarity joins using mapreduce. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, SIGMOD '10, pages 495–506, New York, NY, USA, 2010. ACM. 20

[VME03] V. S. Verykios, G. V. Moustakides, and M. G. Elfeky. A bayesian decision model for cost optimal record matching. *The VLDB Journal*, 12(1):28–40, 2003. 54

[WC10] Meng-Sung Wu and Jen-Tzung Chien. A new topic-bridged model for transfer learning. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 5346 –5349, march 2010. 29, 107

[WeP09] WePS2. Second Web People Search Evaluation Workshop, WePS-2-dataset. http://nlp.uned.es/weps/weps-2-data/, 2009. 48

[Whi09] Tom White. *Hadoop: The Definitive Guide*. O'Reilly Media, 2009. 20, 114, 120

[WHMW11] Shaomei Wu, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts. Who says what to whom on twitter. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 705–714, New York, NY, USA, 2011. ACM. 23

[Win99] W. Winkler. The state of record linkage and current research problems. 1999. 14, 15

[WKB97] Kevin Woods, W. Philip Kegelmeyer, Jr., and Kevin Bowyer. Combination of multiple classifiers using local accuracy estimates. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(4):405–410, 1997. 23, 55

[WKFF12] Jiannan Wang, Tim Kraska, Michael J. Franklin, and Jianhua Feng. Crowder: crowdsourcing entity resolution. *Proc. VLDB Endow.*, 5(11):1483–1494, July 2012. 17

[WLJH10] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 261–270, New York, NY, USA, 2010. ACM. 27, 28, 105

[WMK$^+$09] Steven Euijong Whang, David Menestrina, Georgia Koutrika, Martin Theobald, and Hector Garcia-Molina. Entity resolution with iterative blocking. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, SIGMOD '09, pages 219–232, New York, NY, USA, 2009. ACM. 20

[WWP06] William E. Winkler, William E Winkler, and Nov P. Overview of record linkage and current research directions. Technical report, Bureau of the Census, 2006. 16

[WYLL11] Jianshu Weng, Yuxia Yao, Erwin Leonardi, and Francis Lee. Event detection in twitter. Technical report, HP Labs, 2011. 116

[YCDA13] Surender Reddy Yerva, Michele Catasta, Gianluca Demartini, and Karl Aberer. Enhanced named entity disambiguation in tweets leveraging user profiles. In *IEEE 13th International Conference on Information Reuse & Integration, IRI-2013*, August 2013. 130

[YDT06] Qiong Yang, Xiaoqing Ding, and Xiaoou Tang. Incorporating generic learning to design discriminative classifier adaptable for unknown subject in face verification. *Computer Vision and Pattern Recognition Workshop*, 0:32, 2006. 83

[YGTA13] Surender Reddy Yerva, Flavia Adelina Grosan, Alexandru Octavian Tandrau, and Karl Aberer. TripEneer: User-based Travel Plan Recommendation Application. In *7th International AAAI Conference on Weblogs and Social Media*, 2013. 130

[YJA12] Surender Reddy Yerva, Ho Young Jeung, and Karl Aberer. Cloud based Social and Sensor Data Fusion. In *FUSION*, 2012. 130

[YMA10a] Surender Reddy Yerva, Zoltan Miklos, and Karl Aberer. It was easy, when apples and blackberries were only fruits. In *Third WePS Evaluation Workshop: Searching Information about Entities in the Web, CLEF (Notebook Papers/LABs/Workshops)*, 2010. 27, 65, 84, 129

[YMA10b] Surender Reddy Yerva, Zoltán Miklós, and Karl Aberer. Towards better entity resolution techniques for Web document collections. In *1st International Workshop on Data Engineering meets the Semantic Web (DESWeb'2010) (co-located with ICDE'2010)*, 2010. 26, 83, 129

[YMA11] Surender Reddy Yerva, Zoltan Miklos, and Karl Aberer. What have fruits to do with technology? The case of Orange, Blackberry and Apple. In *International Conference on Web Intelligence, Mining and Semantics (WIMS 2011)*, page 48. ACM, 2011. 27, 85, 97, 104, 107, 116, 130

[YMA12a] Surender Reddy Yerva, Zoltan Miklos, and Karl Aberer. Entity-based classification of twitter messages. *International Journal of Computer Science and Applications*, 2012. 104, 107, 108, 130

[YMA12b] Surender Reddy Yerva, Zoltán Miklós, and Karl Aberer. Quality-aware similarity assessment for entity matching in Web data. *Information Systems (to appear)*, 2012. 85, 129

[YMG$^+$12] Surender Reddy Yerva, Zoltán Miklós, Flavia Adelina Grosan, Octavian Alexandru Bivolaru, and Karl Aberer. TweetSpector: Entity-based retrieval of Tweets. In *SIGIR*, 2012. 130

[YMH$^+$] L. Yang, M. Moshtaghi, B. Han, S. Karunasekera, R. Kotagiri, T. Baldwin, and A. Harwood. Mining micro-blogs: Opportunities and challenges. Social Networks: Computational Aspects and Mining. Springer. 29, 106

[YMO$^+$10] Minoru Yoshida, Shin Matsushima, Shingo Ono, Issei Sato, and Hiroshi Nakagawa. ITC-UT: Tweet Categorization by Query Categorization for On-line Reputation Management. In Martin Braschler, Donna Harman, and Emanuele Pianta, editors, *CLEF (Notebook Papers/LABs/Workshops)*, 2010. 26, 74, 84

[YSJA12] Surender Reddy Yerva, Jonnahtan Saltarin, Ho Young Jeung, and Karl Aberer. Social and Sensor Data Fusion in the Cloud. 2012. 130

[YSL11] A. Hanjalic Y. Shi, P. Serdyukov and M. Larson. Personalized landmark recommendation based on geotags from photo sharing sites. *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, 2011. 108

[ZJW$^+$11] Wayne X. Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee P. Lim, Hongfei Yan, and Xiaoming Li. Comparing twitter and traditional media using topic models. In *Proceedings of the 33rd European conference on Advances in information retrieval*, ECIR'11, pages 338–349, Berlin, Heidelberg, 2011. Springer-Verlag. 23, 27, 28, 105, 106

[Zob96] Justin Zobel. Phonetic string matching: Lessons from information retrieval. 1996. 14

[ZR05] Huimin Zhao and Sudha Ram. Entity identification for heterogeneous database integration: a multiple classifier system approach and empirical evaluation. *Information Systems*, 30(2):119–132, April 2005. 23, 56

[ZTL07] Jing Zhang, Jie Tang, and Juanzi Li. Expert finding in a social network. In *Advances in Databases: Concepts, Systems and Applications*, volume 4443 of *Lecture Notes in Computer Science*, pages 1066–1069. Springer Berlin / Heidelberg, 2007. 27, 28, 105, 106

Surender Reddy Yerva

CONTACT INFORMATION

Rte Cantonale 37,Studio 18
1025 St-Sulpice,Switzerland
mobile: +41 78 6281059

twitter: *@imph0enix*
e-mail: *suren82@gmail.com*
http://lsirpeople.epfl.ch/yerva

RESEARCH INTERESTS

Information Retrieval, Machine Learning, Web Data Mining, Large Scale Systems, Cloud Computing, Social Networks, P2P Networks.

EDUCATION

PhD., Computer Science **EPFL, Switzerland** **2008 – 2013**

Thesis Title: "*Entities on the Web: Resolution, Matching and Profiling*".
Advisor: Prof. Karl Aberer

MS in Computer and Communication Sciences, **EPFL, Switzerland** **2005 – 2007**
BTech in Electrical Engineering, **IIT-Madras** **1999 – 2003**

PROFESSIONAL EXPERIENCE

EPFL, Switzerland, *[Teaching Assistant, Research Assistant]* **2008 – 2013**

- Teaching assistant for graduate course : *Distributed Information Systems* for over three years.
- Teaching assistant for undergrad course : *Informatique - 1 (Java).*
- Research contribution to various European projects: ***NisB***, ***Okkam***, ***Nepomuk***.
- Contribution to various open source projects: ***pgrid, gridvine, gsn, tweetspector***.
- Guided several bachelor and master students with their semester projects.

IBM Research Labs, Zurich, Switzerland, *[MS Thesis Student]* **2006 – 2007**

Multihop Network Simulations: Performance Evaluation of a novel routing algorithm for sensor networks which makes use of diversity combining. This project involved modeling of various protocol layers in an extensible manner. It required development of models in C++, OMNeT++ and matlab.

Qwest Software Services, Bangalore, India, *[Software Engineer]* **2004 – 2005**

Performance Tuning: With good knowledge of Distributed Information Systems, .Net Framework and Database Management skills was able to fine tune an application, and reduce the response time from over 120 seconds to less than 3 seconds. Work load analysis, application tuning and fragmentation enabled to achieve the significant performance gains.

CTS, Chennai, India, *[Software Engineer]* **2003 – 2004**

Library Management System: Lead a team of five in developing LMS software, which aids the librarian to maintain library effectively and easily.

PROGRAMMING

Java, Python, C#, C++, C, Hadoop, NoSQL, Matlab, Linux shell scripting, Perl, Databases, .NET.

SCHOLASTIC ACHIEVEMENTS

Our classification system had the most accurate results for the WePS-3 Challenge-2, Sept 2010

Scholarship: Master students scholarship awarded by EPFL based on academic excellence.2005-2007

One of the best IT Teams for year 2004 at Qwest Communications for Qcare Project., January 2005

All India Rank of 292 from over 150,000 students in IIT-JEE 1999. (Top 0.3 % at national level), 1999

PUBLICATIONS

Surender Yerva, Michele Catasta, Gianluca Demartini, Karl Aberer. "*Entity disambiguation in Tweets leveraging user Social Profiles*". IEEE IRI, 2013, SFO, USA. August 2013.

Surender Yerva, Flavia Grosan, Alexandru Tandrau, Karl Aberer. "*TripEneer: User-based Travel Plan Recommendation Application*". ICWSM, 2013, Boston, USA. July 2013.

Surender Yerva, Zoltan Miklos, Flavia Grosan, Alexandru Tandrau, Karl Aberer. "*TweetSpector: Entity-based Retrieval of Twitter Messages*". SIGIR, 2012, Portland, USA. August 2012.

Surender Yerva, Hoyoung Jeung, Karl Aberer. “*Cloud based fusion of Social and Sensor Data*”. FUSION, 2012, Singapore. July 2012.

Surender Yerva, Jonnahtan Saltarin, Hoyoung Jeung, Karl Aberer. “*Social and Sensor Data Fusion in the Cloud*”. MDM 2012, Bengaluru. July 2012.

Surender Yerva, Zoltan Miklos, Karl Aberer. “*Entity-based Classification of Twitter Messages*”. International Journal of Computer Science and Applications. February 2012.

Surender Yerva, Zoltan Miklos, Karl Aberer. “*Quality-aware similarity assessment for entity matching in Web data*”. Information Systems Journal. September 2011.

Surender Yerva, Zoltan Miklos, Karl Aberer. “*What have fruits to do with technology? The case of Orange, Blackberry and Apple*”. WIMS’11, Norway. May 2011.

Surender Yerva, Zoltan Miklos, Karl Aberer. “*It was easy, when apples and blackberries were only fruits*”. WePS-3, CLEF 2010, Paduva, Italy.2010.

Surender Yerva, Zoltan Miklos, Karl Aberer. “*Towards Better Entity Resolution Techniques for Web Document Collections*”.DesWEB, ICDE-2010, Long Beach, USA. March 2010.

Distributed Components in NEPOMUK social semantic desktop and Prototyping distributed applications, NEPOMUK Summer School Tutorial 2008, Malta.[Tutorial]

Memberships

External Reviewer:

VLDB 2012, BPM 2012, EDBT 2012, ESWC 2011, P2P 2011, EDBT 2011, MDM 2010, ICWS 2010, WebDB 2010, Middleware 2010, MDM 2009, ODBASE 2009, P2P 2008, SIGMOD 2008, VLDB 2008

Program Committee:

Fifth International Conference on Social Informatics (http://www.socinfo2013.org/), Nov 2013, Kyoto

Web Chair:

Fourth International Conference on Social Informatics (http://www.socinfo2012.com/), Dec 2012, Lausanne

Student Member:

YUVA: Indian Students Association Member & Volunteer: 2010-*present*

Sports Committee Member:

Qwest Software Services, Bangalore, 2004 - 2005

Languages

English (Fluent), French (Basic), Hindi (Fluent), Telugu (Native)